



# Computing the double-gyroaverage term incorporating short-scale perturbation and steep equilibrium profile by the interpolation algorithm

Shuangxi Zhang, Michel Mehrenberger, Christophe Steiner

## ► To cite this version:

Shuangxi Zhang, Michel Mehrenberger, Christophe Steiner. Computing the double-gyroaverage term incorporating short-scale perturbation and steep equilibrium profile by the interpolation algorithm. *plasma*, 2019, 2 (2), pp.91-126. 10.3390/plasma2020009 . hal-02009446v2

**HAL Id: hal-02009446**


**<https://hal.science/hal-02009446v2>**

Submitted on 1 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Computing the Double-Gyroaverage Term Incorporating Short-Scale Perturbation and Steep Equilibrium Profile by the Interpolation Algorithm

Shuangxi Zhang <sup>1,\*</sup>, Michel Mehrenberger <sup>2</sup>  and Christophe Steiner <sup>1</sup>

<sup>1</sup> Institut de Recherche Mathématique Avancée, 7 rue René Descartes, 67084 Strasbourg CEDEX & Inria TONUS Team, France; christophe.steiner1@gmail.com

<sup>2</sup> Aix Marseille Univ, CNRS, Centrale Marseille, I2M, Marseille, France; michel.MEHRENBARGER@univ-amu.fr

\* Correspondence: zshuangxi@gmail.com

Received: 6 February 2019; Accepted: 3 April 2019; Published: 9 April 2019



**Abstract:** In the gyrokinetic model and simulations, when the double-gyroaverage term incorporates the combining effect contributed by the finite Larmor radius, short scales of the perturbation, and steep gradient of the equilibrium profile, the low-order approximation of this term could generate unignorable error. This paper implements an interpolation algorithm to compute the double-gyroaverage term without low-order approximation to avoid this error. For a steep equilibrium density, the obvious difference between the density on the gyrocenter coordinate frame and the one on the particle coordinate frame should be accounted for in the quasi-neutrality equation. A Euler–Maclaurin-based quadrature integrating algorithm is developed to compute the quadrature integral for the distribution of the magnetic moment. The application of the interpolation algorithm to computing the double-gyroaverage term and to solving the quasi-neutrality equation is benchmarked by comparing the numerical results with the known analytical solutions. Finally, to take advantage of the interpolation solver clearer, the numerical comparison between the interpolation solver and a classical second order solver is carried out in a constant theta-pinch magnetic field configuration using SELALIB code. When the equilibrium profile is not steep and the perturbation only has the non-zero mode number along the parallel spatial dimension, the results computed by the two solvers match each other well. When the gradient of the equilibrium profile is steep, the interpolation solver provides a bigger driving effect for the ion-temperature-gradient modes, which possess large polar mode numbers.

**Keywords:** electrostatic gyrokinetic model; double-gyroaverage term; interpolation algorithm; euler–maclaurin quadrature integrating formula

## 1. Introduction

Micro-scale turbulence plays a significant role in the confinement capability of magnetized fusion plasma through its interaction with low-frequency zonal flow [1–7], equilibrium profile at the pedestal region, and edge localized modes [8–12], etc. The importance of the pedestal region is reflected by the fact that the pressure of the plasma core is proportional to the pressure at the pedestal top [13–15]. The gradient of the equilibrium temperature and density at the pedestal region could be very sharp; alternatively, the truncation of  $\exp(\rho_0 \cdot \nabla)$  acting over  $n_0$  or  $T_0$  at the first order, which is used by the standard gyrokinetic model [16–18], is not a good approximation, where  $\rho_0$  is the Larmor radius vector defined as

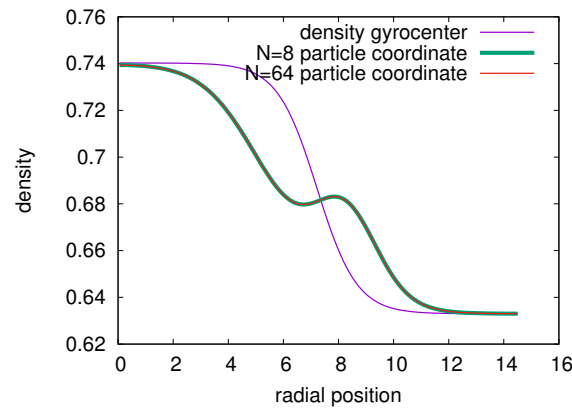
$$\rho_0(\mathbf{x}, \mu, \theta) = \frac{1}{q_s} \sqrt{\frac{2\mu m_s}{B}} (\mathbf{e}_1 \cos \theta + \mathbf{e}_2 \sin \theta).$$

Here,  $q_s, m_s, \mu$  are the particles' charge, mass, and the magnetic moment on the guiding-center coordinate frame;  $\mathbf{e}_1, \mathbf{e}_2$  are the unit vectors perpendicular to unit vector  $\mathbf{b}$  of the equilibrium magnetic field, and  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{b}$  obey the right-hand rule;  $B$  is the magnitude of the magnetic field;  $\theta$  is the gyrophase angle;  $n_0, T_0$  are the equilibrium density and temperature profile. In gyrokinetic theory, the first gyroaverage term could incorporate the effect contributed by the finite Larmor radius and short scales of the perturbation at the core and edge tokamak plasma, while the double-gyroaverage term (DGT) incorporates the combining effect contributed by the finite Larmor radius, the short-scale perturbation and the steep-gradient equilibrium profile within the edge transport barrier of tokamak plasma [18–21]. However, in the standard electrostatic gyrokinetic model and simulations, the approximation of DGT consists of up to the first order truncation of the Taylor expansion of the equilibrium profile and the second order truncation of the expansion of the perturbative potential [16,17,22–28]. The details of this method can be found in Appendix A. These kinds of low-order approximation are not sufficient concerning the short-scale perturbations and the equilibrium profile possessing the steep gradient.

To overcome the mentioned drawbacks of the low-order approximation of DGT, this paper develops an interpolation algorithm to compute DGT for the purpose of the resolving of the short-scale perturbation and the steep equilibrium profile together. The interpolation algorithm can approach DGT with arbitrary accuracy by choosing enough sampling points on the Larmor circle, with only the constraint from the length scale of the mesh. Meanwhile, contrary to the traditional way, the steep equilibrium density profile should not keep the same before and after the gyroaverage operation, which is denoted by the symbol  $\mathcal{J}$  in this paper. And the obvious difference between the density on gyrocenter coordinate and the one on the particle coordinates should be accounted for in the quasi-neutrality equation(QNE). The obvious difference is revealed by Figure 1, for which the normalization scheme can be found in Section 3.3 and the parameters to obtain this figure are provided in Section 7.3.3. In Figure 1, the purple line denotes the equilibrium density profile on the gyrocenter coordinates, which is given as the initial condition, while the density profiles on the particle coordinate derived by the gyroaverage operation computed with different number of  $\mu$  are shown by other curves. The numerical integration of  $\mu$  is done by the Euler–Maclaurin-based quadrature integration algorithm.

As a comparison to the interpolation solver, a classical 2nd order truncation of DGT is carried out in this paper to obtain QNE with the 2nd-order accuracy. It consists of the truncation of the exponential operator  $\exp(\rho_0 \cdot \nabla)$  over the potential up to the second order and the truncation of the operator  $\exp(\rho_0 \cdot \nabla)$  acting on the density up to the first order. The numerical simulations are carried out to compare the two solvers. Since this paper is to compare the two solvers in general purpose, the deduction of flux-surface average of the potential from the total potential is not carried out to obtain the electrons' adiabatic distribution. For the equilibrium profile without steep gradient, both algorithms have almost the same performance in terms of the perturbation only possessing the mode number along the parallel spatial dimension. For the steep equilibrium profile, the interpolation algorithm could provide a stronger driving effect for the perturbation of high polar mode numbers, and the saturation time of these modes computed by the interpolation algorithm is earlier.

The remaining parts of this paper are arranged as follows. Section 2 introduces the orders used in deriving the gyrokinetic model. The DGT and QNE incorporating the short-scale perturbation and steep equilibrium profile on the gyrocenter coordinate frame are derived in Section 3. In Section 4, the interpolation solver is introduced. The benchmark of the interpolation algorithm is done in Sections 5 and 6 for the case of the single  $\mu$  and  $\mu$  obeying the distribution, respectively. The Euler–Maclaurin-based quadrature integrating algorithm is developed in Section 6 to compute the numerical quadrature integral of  $\mu$ . The application of the interpolation algorithm and the Euler–Maclaurin-based quadrature integrating algorithm to the gyrokinetic simulations is provided by Section 7, where the parallel scheme of the whole simulations is presented.



**Figure 1.** The equilibrium density profile on the gyrocenter coordinate before the gyroaverage operation is shown by the purple line, while the ones on the particle coordinate derived by the gyroaverage operation and with the quadrature integral of  $\mu$  computed by the Euler–Maclaurin-based algorithm with the forward finite difference scheme for  $\mu$  meshes of 8 and 64 nodes are presented by the other two curves. The radial distance is normalized by  $\rho_0$ .

## 2. The Basic Orders

There are several basic orders or scales contained by the perturbation. The first one is the length scale  $O(\varepsilon)$  of the nondimensionalized Larmor radius being  $\varepsilon$ . The second one is the amplitude of the normalized electrostatic potential, whose  $\varepsilon$ -based order is denoted as  $O(\varepsilon^\sigma)$  with the power  $\sigma$  a positive real number. In the magnetically confined fusion plasmas, because the charged particle can nearly migrate freely in the environment with collective interactions, the magnitude of the potential the particles feel must be much smaller than that of its kinetic energy.

The third one is the length scale of the gradient of the electrostatic potential. Define  $\mathcal{K}_\perp \equiv |\frac{\nabla_\perp \phi}{\phi}|$  and  $\mathcal{K}_\parallel \equiv |\frac{\nabla_\parallel \phi}{\phi}|$ , where the subscript  $\perp$  and  $\parallel$  denotes the directions perpendicular and parallel the direction of equilibrium magnetic field, respectively. The gyrokinetic model of this paper adopts the scales for the perturbation

$$O(\varepsilon \mathcal{K}_\perp) = O(1), \quad O(\varepsilon \mathcal{K}_\parallel) = O(\varepsilon), \quad (1)$$

where the short scales of the perturbation are accounted for. For any equilibrium quantity  $\mathcal{E} \in \{n_0, T_0\}$ , this paper considers the scale

$$O\left(\varepsilon \left\| \frac{\nabla_\perp \mathcal{E}}{\mathcal{E}} \right\| \right) = O(1), \quad (2)$$

due to the steep gradient of the equilibrium profiles, as well as the following scales

$$O\left(\varepsilon \left\| \frac{\nabla_\parallel \mathcal{E}}{\mathcal{E}} \right\| \right) = O(\varepsilon), \quad O\left(\varepsilon \left\| \frac{\partial U \mathcal{E}}{\mathcal{E}} \right\| \right) = O(\varepsilon),$$

where  $U$  is the parallel velocity and will be given later.

## 3. QNE Incorporating the Short-Scale Perturbation and Steep Equilibrium Profile

### 3.1. The Particle Coordinate Density Associated with the Short-Scale Perturbation and Steep Equilibrium Profile

The procedure to derive the gyrokinetic model is composited by two parts. The first one is to derive the coordinate transform by decoupling the gyroangle from the dynamics of the remaining variables, while the second one is to obtain the gyrokinetic quasi-neutrality equation by inducing the transformation of the distribution through the derived coordinate transforms [16,19,29,30]. The first step is accomplished by implementing Lie transform perturbation method on the fundamental Lagrangian one-form. Generally, four kinds of coordinate frameworks are involved in the procedure.

The first one is the full-orbit coordinate with the velocity part in Cartesian coordinates. it is denoted as  $\bar{\mathbf{z}} \equiv (\mathbf{x}, \mathbf{v})$  here. The second one is obtained by transforming  $\mathbf{v}$  into the cylindrical coordinates, and it is written as  $\mathbf{z} \equiv (\mathbf{x}, \mu_1, u_1, \theta_1)$  where  $\mu_1 \equiv \frac{mv_1^2}{2B(\mathbf{x})}$ ,  $u_1$  is the velocity along the parallel direction, and  $\theta_1$  is the angle between  $\boldsymbol{\rho}$  and  $\mathbf{e}_1$ . The  $\mathbf{x}$  component in  $\mathbf{z}$  is still in full-orbit frame. The third one is the guiding-center coordinates  $\bar{\mathbf{Z}} = (\bar{\mathbf{X}}, \bar{\mu}, \bar{U}, \bar{\theta})$ , which is derived by decoupling  $\bar{\theta}$  from the dynamics of the other coordinate components without the presence of the perturbation. The fourth one is the gyrocenter coordinate  $\mathbf{Z} = (\mathbf{X}, \mu, U, \theta)$  which is derived by decoupling  $\bar{\theta}$  from the dynamics of the other coordinate components with the presence of the perturbation. The coordinate transforms between  $\bar{\mathbf{z}}, \mathbf{z}, \bar{\mathbf{Z}}$  and  $\mathbf{Z}$  are denoted as  $\psi_f : \bar{\mathbf{z}} \rightarrow \mathbf{z}$ ,  $\psi_{gc} : \mathbf{z} \rightarrow \bar{\mathbf{Z}}$  and  $\psi_{gy} : \bar{\mathbf{Z}} \rightarrow \mathbf{Z}$ , respectively, while the distributions on the four kinds of coordinates are respectively written as  $\bar{f}(\bar{\mathbf{z}})$ ,  $f(\mathbf{z})$ ,  $\bar{F}(\bar{\mathbf{Z}})$  and  $F(\mathbf{Z})$ .

The coordinate transform  $\psi_{gc} : \mathbf{z} \rightarrow \bar{\mathbf{Z}}$  is given by

$$\bar{\mathbf{X}} = \mathbf{x} - \rho_0(\mathbf{x}, \mu_1, \theta_1), \quad (3a)$$

$$\bar{\mu} = \mu_1, \quad (3b)$$

$$\bar{U} = u_1, \quad (3c)$$

$$\bar{\theta} = \theta_1. \quad (3d)$$

while the coordinate transform  $\psi_{gy} : \bar{\mathbf{Z}} \rightarrow \mathbf{Z}$  is provided by the following equations

$$\mathbf{X} = \bar{\mathbf{X}}, \quad (4a)$$

$$\mu = \bar{\mu} + g_2^\mu(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}), \quad (4b)$$

$$U = \bar{U}, \quad (4c)$$

$$\theta = \bar{\theta} + g_2^\theta(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}). \quad (4d)$$

where

$$g_2^\mu(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) = \frac{q_s \Psi(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta})}{B(\bar{\mathbf{X}})}, \quad (5a)$$

$$g_2^\theta(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) = \frac{q_s \partial_{\bar{\mu}} \int_0^{\bar{\theta}} \Psi(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) d\bar{\theta}}{B(\bar{\mathbf{X}})}, \quad (5b)$$

and

$$\Psi(\bar{\mathbf{X}}, \bar{\mu}) \equiv \phi(\bar{\mathbf{X}} + \rho_0) - \Phi(\bar{\mathbf{X}}, \bar{\mu}), \quad (6a)$$

$$\Phi(\bar{\mathbf{X}}, \bar{\mu}) \equiv \frac{1}{2\pi} \int_0^{2\pi} \phi(\bar{\mathbf{X}} + \rho_0) d\theta, \quad (6b)$$

$$\rho_0(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}) = \frac{1}{q_s} \sqrt{\frac{2\bar{\mu}m_s}{B}} (\mathbf{e}_1 \cos \bar{\theta} + \mathbf{e}_2 \sin \bar{\theta}). \quad (6c)$$

And through the Lie transform perturbative method, we could derive the following Lagrangian on the gyrocenter coordinate frame

$$\mathcal{L} = (q_s \mathbf{A}(\mathbf{X}) + m_s U \mathbf{b}) \cdot \dot{\mathbf{X}} + \frac{m_s}{q_s} \mu \dot{\theta} - \left( \mu B(\mathbf{X}) + \frac{m_s U^2}{2} + \Phi(\mathbf{X}, \mu) \right), \quad (7)$$

where  $\mu$  is a constant. The Euler-Lagrangian equations based on  $\mathcal{L}$  in Equation (7) provides the equations of motion as follows:

$$\dot{\mathbf{X}} = \frac{q_s U \mathbf{B}^* - \mathbf{b} \times \nabla (\mu B + q_s \Phi(\mathbf{X}, \mu))}{q_s B_{\parallel}^*}, \quad (8a)$$

$$\dot{U} = \frac{\mathbf{B}^* \cdot \nabla (\mu B + q_s \Phi(\mathbf{X}, \mu))}{m_s B_{\parallel}^*}, \quad (8b)$$

where  $\mathbf{B}^* = \mathbf{B} + \frac{m_s}{q_s} U \nabla \times \mathbf{b}$ . If the denominator in Equation (8a) is expanded by the order of the small factor  $\frac{m_s}{q_s} U \nabla \times \mathbf{b}$ , we could obtain the curvature drift part  $\frac{m U^2 \kappa}{e B}$  where  $\kappa = \frac{\nabla B}{B}$ .

For the Vlasov gyrokinetic simulation, we need to transform the distribution function from the gyrocenter coordinate to the full-orbit coordinate [16]. After obtaining the coordinate transform composited by Equations (3a)–(3d) and (4a)–(4d), given a distribution function on the gyrocenter coordinate  $F_s(\mathbf{X}, \mu, U, t)$ , the distribution function on the full orbit can be derived following the transform chain

$$F_s(\mathbf{X}, \mu, U) \xrightarrow{\psi_{gy}} \bar{F}_s(\bar{\mathbf{Z}}) \xrightarrow{\psi_{gc}} f_s(\mathbf{z}). \quad (9)$$

First, the total distribution function is separated into the sum of an equilibrium one plus a perturbative one as

$$F_s(\mathbf{X}, \mu, U) = F_{s0}(\mathbf{X}, \mu, U) + F_{s1}(\mathbf{X}, \mu, U). \quad (10)$$

Then, the approximation of the distribution on the guiding-center coordinate can be derived based on the coordinate transform given by Equations (4a)–(4d)

$$\begin{aligned} \bar{F}_s(\bar{\mathbf{Z}}) &= F_s(\bar{\mathbf{X}}, \bar{\mu} + g_2^\mu(\bar{\mathbf{X}}, \bar{\mu}, \bar{\theta}), \bar{U}) \\ &\approx F_s(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}) + \frac{q_s \Psi(\bar{\mathbf{Z}})}{B(\bar{\mathbf{Z}})} \partial_{\bar{\mu}} F_{s0}(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}) \\ &= F_s(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}) - \frac{q_s \Psi(\bar{\mathbf{Z}})}{T_s(\bar{\mathbf{Z}})} F_{s0}(\bar{\mathbf{X}}, \bar{\mu}, \bar{U}), \end{aligned} \quad (11)$$

where the equilibrium distribution involving  $\mu$  is assumed as

$$F_{s0\perp} = \frac{m_s}{2\pi T_s} \exp\left(-\frac{\mu B}{T_s}\right). \quad (12)$$

The approximation of  $f_s(\mathbf{z})$  is derived with Equation (11) as the base

$$\begin{aligned} f_s(\mathbf{z}) &= F_s(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1) - q_s \frac{\phi(\mathbf{x}) F_{s0}(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1)}{T_s(\mathbf{x})} \\ &\quad + q_s \left( \frac{\Phi F_{s0}}{T_s} \right) (\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1). \end{aligned} \quad (13)$$

where  $\left( \frac{\Phi F_{s0}}{T_s} \right) (\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{x}, \mu_1, \theta_1), \mu_1, u_1)$  is the origin of DGT and comprises the effect combining together the finite Larmor radius, short-scale perturbation, and steep equilibrium profile.

We assume that the equilibrium distribution  $F_{s0}$  can be decomposed as the product between the parallel part and the perpendicular part

$$F_{s0}(\mathbf{x}, \mu_1, u_1) = n_0(\mathbf{x}) F_{s0\parallel}(\mathbf{x}, u_1) F_{s0\perp}(\mathbf{x}, \mu_1), \quad (14)$$

with probability conservation satisfied by

$$\int F_{s0\parallel} du_1 = 1, \quad (15a)$$

$$\int F_{s0\perp} \frac{B(\mathbf{x})}{m_s} d\mu_1 d\theta_1 = 1, \quad (15b)$$

where under the equilibrium condition, the metric  $B(\mathbf{x})/m_s$  is used.

Then, through the integral  $n_s(\mathbf{x}, t) = \int f_s(\mathbf{z}) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1$ , the density can be assembled as

$$n_s(\mathbf{x}) = n_{sg0}(\mathbf{x}) - q_s \frac{n_{sg0}\phi}{T_s} + q_s \tilde{\phi}'(\mathbf{x}) + n_{sg1}(\mathbf{x}, t), \quad (16)$$

with

$$n_{sg0}(\mathbf{x}) = \int F_{s0}(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{z}), \mu_1, u_1) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1, \quad (17a)$$

$$n_{sg1}(\mathbf{x}, t) = \int F_{s1}(\mathbf{x} - \boldsymbol{\rho}_0(\mathbf{z}), \mu_1, u_1) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1, \quad (17b)$$

$$\tilde{\phi}'(\mathbf{x}) = \int \left( \frac{\Phi F_{s0}}{T_s} \right) (\mathbf{x} - \boldsymbol{\rho}_0, \mu_1, u_1) \frac{B(\mathbf{x})}{m_s} d\mu_1 du_1 d\theta_1. \quad (17c)$$

Here, the metric  $B(\mathbf{x})/m_s$  of the phase space is used.

### 3.2. QNE Associated with the Adiabatic Distribution of Electron and the Steep Equilibrium Profile on Gyrocenter Coordinate Frame

As shown in Figure 1, the equilibrium density profile possessing the steep gradient on the gyrocenter coordinate frame is obviously different from the one on the particle coordinate frame after the gyroaverage operation. Therefore, contrary to the traditional way, the implementation of the equilibrium density on the particle coordinate frame should be different from the one on the gyrocenter coordinate frame if its profile is steep in the radial dimension.

We consider a plasma only including protons and electrons. The electrons obey the adiabatic distribution on the particle coordinate space

$$n_e(\mathbf{x}) = n_{g0}(\mathbf{x}) + \frac{en_{g0}(\mathbf{x})}{T_e} \phi(\mathbf{x}), \quad (18)$$

where the flux-surface average of the potential is not deducted from the total potential, because our purpose is to compare the two solvers.  $n_{g0}(\mathbf{x})$  is the equilibrium density in the particle coordinate space and needs to be solved. The initial equilibrium distribution of ion is given on the guiding-center coordinate system by

$$F_{i0}(\mathbf{x}, \mu, U) = \frac{n_0(\mathbf{x})}{(2\pi T_i(\mathbf{x}))^{3/2}} \exp\left(-\frac{U^2}{2T_i(\mathbf{x})} - \frac{\mu B(\mathbf{x})}{T_i(\mathbf{x})}\right),$$

from which the equilibrium density on the guiding-center coordinate is easily derived as  $n_0(\mathbf{x})$ . For the scales given by Equation (2), the equilibrium density on the particle coordinate space should be derived as

$$n'_0 = \int F_{i0}(\mathbf{x} - \boldsymbol{\rho}_0, \mu, U) \frac{B}{m_i} d\mu_1 dU d\theta \quad (19)$$

Taking into account of the quasi-neutrality equilibrium, it is derived that

$$n_{g0}(\mathbf{x}) = n'_0(\mathbf{x}). \quad (20)$$

Then, QNE of this plasma is

$$\frac{e\phi}{T_i} - \frac{e\tilde{\phi}'}{n_{g0}} + \frac{e\phi}{T_e} = \frac{n_{i1}}{n_{g0}}. \quad (21)$$

### 3.3. Normalization

In this paper, the quantities  $t, v, B, l, \mu, T, \phi$  are normalized by  $t_0 \equiv \frac{m}{B_0 q_i}, v_0 \equiv \sqrt{\frac{T_{e0}}{m_i}}, B_0, l_0 \equiv \frac{mv_0}{eB_0}, \mu_0 \equiv \frac{T_{e0}}{B_0}, T_{e0}$  and  $\phi_0 \equiv \frac{T_{e0}}{q_i}$ , respectively, where  $T_{e0} \equiv T_e(r_p)$  and  $r_p \in [r_{min}, r_{max}]$  is the radial position of the peak of the initial distribution function. Then, by choosing the equilibrium perpendicular distribution given by Equation (12), the normalized QNE is

$$\frac{\phi(\mathbf{x})}{T_i(\mathbf{x})} - \frac{\tilde{\Phi}(\mathbf{x})}{n_{g0}(\mathbf{x})} + \frac{\phi(\mathbf{x})}{T_e(\mathbf{x})} = \frac{n_{i1}(\mathbf{x})}{n_{g0}(\mathbf{x})}, \quad (22)$$

with the normalized DGT

$$\tilde{\Phi}(\mathbf{x}) = \int \left( \frac{\Phi F_{i0}}{T_i} \right) (\mathbf{x} - \rho_0, \mu_1, u_1) B(\mathbf{x}) d\mu_1 du_1 d\theta_1, \quad (23)$$

$$\text{and } F_0 = \frac{n_0(r) \exp\left(-\frac{U^2}{2T_i(r)} - \frac{\mu B}{T_i(r)}\right)}{(2\pi T_i(r))^{3/2}}.$$

## 4. QNE Solver Comprising the Interpolation Algorithm to Compute DGT

### 4.1. The Algorithm to Compute the Interpolation Matrixes $A^{spl}$ and $A_{l,\rho_j}^{contr}$ in Finite Element Method

In this paper, we sometimes use the symbols  $\mathcal{J}_1(\mu)$  and  $\mathcal{J}_2(\mu)$  to denote the first and the second gyroaverage associated with the magnetic moment  $\mu$ , specifically,

$$\mathcal{J}_1(\mu)\phi(\mathbf{x}) = \frac{1}{2\pi} \int \phi(\mathbf{x} + \rho_0(\mu)) d\theta,$$

$$\mathcal{J}_2(\mu)\phi(\mathbf{x}) = \frac{1}{2\pi} \int \phi(\mathbf{x} - \rho_0(\mu)) d\theta.$$

In our interpolation scheme of the finite element method(FEM), the following cubic spline function is used as the basis function

$$B_\alpha(x) = \frac{1}{6h^3} \begin{cases} (x - x_{\alpha-2})^3 & \text{if } x_{\alpha-2} \leq x \leq x_{\alpha-1} \\ h^3 + 3h^2(x - x_{\alpha-1}) + 3h(x - x_{\alpha-1})^2 - 3(x - x_{\alpha-1})^3 & \text{if } x_{\alpha-1} \leq x \leq x_\alpha \\ h^3 + 3h^2(x_{\alpha+1} - x) + 3h(x_{\alpha+1} - x)^2 - 3(x_{\alpha+1} - x)^3 & \text{if } x_\alpha \leq x \leq x_{\alpha+1} \\ (x_{\alpha+2} - x)^3 & \text{if } x_{\alpha+1} \leq x \leq x_{\alpha+2} \\ 0 & \text{otherwise} \end{cases}$$

The spatial domains  $(x, y, z)$ , where  $z$  is the coordinate parallel to the equilibrium magnetic unit vector, the parallel velocity  $U$  and the magnetic moment  $\mu$  are equally discretized.

To obtain the matrix form, the physical quantities on the grid mesh are written in the vectors as follows:



$$\begin{aligned}
 \psi_{h,p} &:= \psi(x_h, y_p), h \in \{0, \dots, N_x\}; p \in \{0, \dots, N_y - 1\} \\
 \psi_{h,p,j} &:= \psi(x_h, y_p, \mu_j), j \in \{1, \dots, N_\mu\} \\
 \psi_{h,p,jk} &:= \psi(x_h, y_p, \mu_j, U_k), k \in \{1, \dots, N_U\} \\
 \phi &:= (\phi_{0,0}, \dots, \phi_{N_x,0}, \phi_{0,1}, \dots, \phi_{N_x,1}, \dots, \phi_{0,N_y-1}, \dots, \phi_{N_x,N_y-1})^t, \\
 \Phi_j &:= (\Phi_{0,0,j}, \dots, \Phi_{N_x,0,j}, \Phi_{0,1,j}, \dots, \Phi_{N_x,1,j}, \dots, \Phi_{0,N_y-1,j}, \dots, \Phi_{N_x,N_y-1,j})^t, \\
 \mathcal{F}_{sjk} &:= (\mathcal{F}_{0,0,jk}, \dots, \mathcal{F}_{N_x,0,jk}, \mathcal{F}_{0,1,jk}, \dots, \mathcal{F}_{N_x,1,jk}, \dots, \mathcal{F}_{0,N_y-1,jk}, \dots, \mathcal{F}_{N_x,N_y-1,jk})^t, \\
 \tilde{\Phi}_{jk} &:= (\tilde{\Phi}_{0,0,jk}, \dots, \tilde{\Phi}_{N_x,0,jk}, \tilde{\Phi}_{0,1,jk}, \dots, \tilde{\Phi}_{N_x,1,jk}, \dots, \tilde{\Phi}_{0,N_y-1,jk}, \dots, \tilde{\Phi}_{N_x,N_y-1,jk})^t, \\
 \tilde{\Phi}_{jk} &:= (\tilde{\Phi}_{0,0,jk}, \dots, \tilde{\Phi}_{N_x,0,jk}, \tilde{\Phi}_{0,1,jk}, \dots, \tilde{\Phi}_{N_x,1,jk}, \dots, \tilde{\Phi}_{0,N_y-1,jk}, \dots, \tilde{\Phi}_{N_x,N_y-1,jk})^t.
 \end{aligned}$$

Here,  $\phi$  and  $\Phi$  are the electrostatic potential and the first gyroaverage of the potential, respectively. In addition, the definition of  $\mathcal{F}_s$ ,  $\tilde{\Phi}$  and  $\tilde{\Phi}$  will be given in the subsequent context.

In the finite element method with the cubic spline basis, the potential on the grid point  $(h, p)$  is

$$\phi_{h,p} = \sum_{a,b=-1}^2 C_{h+a,p+b} B_{h+a}(x_h) B_{p+b}(y_p), \quad (24)$$

where  $x$  and  $y$  can be the coordinates in the Cartesian frame or in the polar frame.  $C_{h+a,p+b}$  is the weight of the two-dimensional basis  $B_{h+a} B_{p+b}$ .  $B_{h+a}(x_h) / B_{p+b}(y_p)$  is the value of the one dimensional basis  $B_{h+a} / B_{p+b}$  at the node  $x_h / y_p$ . By assembling  $\phi_{h,p}$  into the vector  $\phi$  as explained before, Equation (24) can be rewritten as the tensor product of two matrixes multiplying the vector  $C$  comprising the weight coefficients of all the bases

$$\phi = \mathcal{B}_b \otimes \mathcal{B}_p C, \quad (25)$$

where

$$C \equiv (C_{0,0}, \dots, C_{N_x,0}, C_{0,1}, \dots, C_{N_x,1}, \dots, C_{0,N_y-1}, \dots, C_{N_x,N_y-1})^t,$$

and the subscript  $P$  denotes the periodic boundary condition used in the  $y$  dimension, while  $b$  denotes the periodic or natural boundary condition used in  $x$  dimension.  $\mathcal{B}_b$  and  $\mathcal{B}_p$  are the matrixes representing the value of  $B_h(x)$ ,  $h \in \{0, 1, 2, \dots, N_x\}$  and  $B_p(y)$ ,  $p \in \{0, 1, \dots, N_y - 1\}$  on the nodes of the  $x$  and  $y$  domains, respectively. Here,  $\mathcal{B}_p$  is of the structure proportional to

$$\begin{bmatrix} 4 & 1 & 0 & \dots & 1 \\ 1 & 4 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ 1 & \dots & 0 & 1 & 4 \end{bmatrix}.$$

Because the rows of  $\mathcal{B}_p(y)$  have the circulant property, the matrix  $\mathcal{B}_p \otimes \mathcal{B}_b$  is of the circulant block structure, too. Its inverse is defined as

$$A^{spl} \equiv (\mathcal{B}_p \otimes \mathcal{B}_b)^{-1} = \mathcal{B}_p^{-1} \otimes \mathcal{B}_b^{-1}.$$

which is still a circulant block matrix and can be written as

$$A^{spl} = \begin{bmatrix} A_0^{spl} & A_1^{spl} & \dots & A_{N_y-1}^{spl} \\ A_{N_y-1}^{spl} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_1^{spl} \\ A_1^{spl} & \dots & A_{N_y-1}^{spl} & A_0^{spl} \end{bmatrix} \in \mathcal{M}_{(N_x+1) \times N_y, (N_x+1) \times N_y}(\mathbb{R}), \quad (26)$$

where  $A_p^{spl}$  is defined as

$$A_p^{spl} = B_p^{-1}(1, p) \mathcal{B}_b^{-1},$$

and  $B_p^{-1}(1, p)$  is the  $(1, p)$ th element of  $\mathcal{B}_p^{-1}$ .

The gyroaverage operations  $\mathcal{J}_{lj}$  with  $l = 1, 2$  on the potential  $\phi(x, y)$  at the node  $(x_h, y_p)$  can be rewritten in the following form

$$\begin{aligned} \mathcal{J}_{lj}\phi(x_h, y_p) &= \frac{1}{N_\theta} \sum_{m=0}^{N_\theta-1} \phi \left( \mathcal{C}_{lj,hpm,1}, \mathcal{C}_{lj,hpm,2} \right) \\ &= \frac{1}{N_\theta} \sum_{m=0}^{N_\theta-1} \sum_{a,b=-1}^2 \mathcal{C}_{lj,D_{lhpm,a},D_{lhpm,b}} val(a, b). \end{aligned} \quad (27)$$

Here, the subscript  $j$  means this gyroaverage is associated with the particles of the magnetic moment  $\mu_j$ , and  $N_\theta$  is the number of the points interpolated on the Larmor circle which surrounds the grid point  $(x_h, y_p)$  and is of the radius  $\rho_j$  equaling  $\sqrt{2\mu_j}$ . The global coordinates of the  $m$ th sampling point on the Larmor circle surrounding the grid point  $(h, p)$  are computed as

$$\mathcal{C}_{lj,hpm,1} = \mathcal{C}_{\min,1} + (d_{lhpm,1} - 1)\delta_1 + c_{lm1}, \quad (28a)$$

$$\mathcal{C}_{lj,hpm,2} = \mathcal{C}_{\min,2} + (d_{lhpm,2} - 1)\delta_2 + c_{lm2}. \quad (28b)$$

In Equations (28a) and (28b),  $\mathcal{C}_{\min,1}$  and  $\mathcal{C}_{\min,2}$  are respectively the minimal value of the  $x$  and  $y$  domain;  $\delta_1$  and  $\delta_2$  are the uniform step length in the respective dimension;  $0 \leq c_{lm1} < \delta_1$  and  $0 \leq c_{lm2} < \delta_2$  hold.  $d_{lhpm,1}$  and  $d_{lhpm,2}$  are the global node numbers associated with the sampling point on the Larmor circle denoted by the indexes  $h, p, m$ .  $\mathcal{C}_{lj,hpm,1}$  and  $\mathcal{C}_{lj,hpm,2}$  can be the global coordinates in the Cartesian or polar coordinate frames. In the 2nd equality of Equation (32), the potential at the sampling point  $(\mathcal{C}_{lj,hpm,1}, \mathcal{C}_{lj,hpm,2})$  is written by the sum of the contribution of the cubic spline basis. We use  $D_{lhpm,a}$  and  $D_{lhpm,b}$  to denote the global node number of the grid point associated with the indexes  $(a, b)$ . This grid point is where the basis function locates. Then,  $val(a, b)$  as the spline function value at  $(\mathcal{C}_{lj,hpm,1}, \mathcal{C}_{lj,hpm,2})$  is given by

$$val(a, b) \equiv B_{D_{lhpm,a}}(c_{m1}) B_{D_{lhpm,b}}(c_{m2}).$$

$\mathcal{C}_{lj,D_{lhpm,a},D_{lhpm,b}}$  in Equation (32) is the weight coefficient of the cubic spline function locating at  $(D_{lhpm,a}, D_{lhpm,b})$ . Eventually, by summing the repeated contribution nodes together, the gyroaverage of  $\phi$  could be written into a matrix form

$$\mathcal{J}_{lj}\phi = A_{l,\rho_j}^{contr} \mathcal{C}_{lj}.$$

Here, we want to mention that due to the finite value of  $\mu_j$ , some sampling points on the Larmor circle of the grid point close to the boundary could locate out of the chosen domain. At the outer boundary, for a realistic plasma, the amplitude of the fluctuation of the potential is close to zero. So we approximate the contribution of such points by the contribution of their projection points on the boundary.

The elements of  $A_{l,\rho_j}^{contr}$  can be denoted as  $A_{l,\rho_j}^{contr}(p_1 + h_1, p_2 + h_2)$  with  $p_1, p_2 \in \{0, \dots, N_y - 1\}$  and  $h_1, h_2 \in \{0, \dots, N_x\}$ . To compute  $A_{l,\rho_j}^{contr}$ , due to its circulant block structure, we only need to compute the block matrixes associated with  $p_1 = 0$ . The other rows of block matrixes associated with  $p_1 \neq 0$  can be obtained by permuting the indexes  $p_2$ s for  $p_1$  times. Specifically,  $A_{l,\rho_j}^{contr}$  is of the following structure

$$A_{l,\rho_j}^{contr} = \begin{bmatrix} A_{l,\rho_j,0}^{contr} & A_{l,\rho_j,1}^{contr} & \dots & A_{l,\rho_j,N_y-1}^{contr} \\ A_{l,\rho_j,N_y-1}^{contr} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{l,\rho_j,1}^{contr} \\ A_{l,\rho_j,1}^{contr} & \dots & A_{l,\rho_j,N_y-1}^{contr} & A_{l,\rho_j,0}^{contr} \end{bmatrix} \in \mathcal{M}_{(N_x+1) \times N_y, (N_x+1) \times N_y}(\mathbb{R}) \quad (29)$$

And the algorithm to compute  $A_{l,\rho_j}^{contr}(h_1, p_2 + h_2)$  is given by Algorithm 1.

---

**Algorithm 1** Compute  $A_{l,\rho_j}^{contr}(h_1, p_2 + h_2)$

---

**Input:**  $\mu_j, l, \delta_1, \delta_2, N_x, N_y, N_\theta$ , **boundary condition**

```

1: for  $h_1 = 0, N_x$  do
2:   for  $m = 1, N_y$  do
3:     get  $d_{lh_1 0m,1}, d_{lh_1 0m,2}$ 
4:     get  $c_1, c_2$ 
5:     get  $val(4, 4)$ 
6:     for  $a = -1, 2$  do
7:       for  $b = -1, 2$  do
8:         get  $D_{lh_1 0ma}, D_{lh_1 0mb}$ 
9:          $A_{l,\rho_j}^{contr}(h_1, D_{lh_1 0ma} + D_{lh_1 0mb}) = A_{l,\rho_j}^{contr}(h_1, D_{lh_1 0ma} + D_{lh_1 0mb}) + val(a, b)$ 

```

---

#### 4.2. Matrix Form of DGT

The polar coordinate system is used for the QNE solver in this paper. To obtain the QNE solver comprising the interpolation algorithm, QNE in Equation (22) is rewritten as

$$\frac{n_{g0}(\mathbf{x})\phi(\mathbf{x})}{T_i(\mathbf{x})} - \tilde{\Phi}(\mathbf{x}) + \frac{n_{g0}(\mathbf{x})\phi(\mathbf{x})}{T_e(\mathbf{x})} = n_{i1}(\mathbf{x}). \quad (30)$$

$\tilde{\Phi}(\mathbf{x})$  can be written as the discrete sum over  $\mu$  and  $U$  as

$$\tilde{\Phi}(\mathbf{x}) = \sum_k \sum_j \tilde{\Phi}(\mathbf{x}, \mu_j, U_k) B \delta \mu_j \delta U_k,$$

where  $\delta \mu_j$  and  $\delta U_k$  are respectively the step length for  $\mu$  and  $U$ . Here,

$$\tilde{\Phi}(\mathbf{x}, \mu_j, U_k) \equiv \frac{1}{2\pi} \int (\Phi \mathcal{F}_s)(\mathbf{x} - \rho_{0j}, \mu_j, U_k) d\theta \quad (31)$$

and in cylindrical coordinate frame,  $\mathcal{F}_s$  is defined as

$$\mathcal{F}_s \equiv \frac{n_0(r) \exp\left(-\frac{U^2}{2T_s(r)} - \frac{\mu B}{T_s(r)}\right)}{(2\pi T_s(r))^{1/2} T_s^2(r)}.$$

The first gyroaverage of  $\phi$  with respect to the magnetic moment  $\mu_j$  is denoted as  $\Phi(\mathbf{x}, \mu_j)$ , whose value at polar mesh node  $(r_h, \Theta_p)$  is written as  $\Phi_{h,p,j}$ . According to Equation (32), it is computed on the grid discretized by the following right Riemann sum:

$$\Phi_{h,p,j} = \frac{1}{N_\theta} \sum_{l=0}^{N_\theta-1} \phi\left(r_h \cos \Theta_p + \rho_j \cos\left(\frac{2\pi l}{N_\theta}\right), r_h \sin \Theta_p + \rho_j \sin\left(\frac{2\pi l}{N_\theta}\right)\right), \quad (32)$$

where  $\rho_j = \sqrt{2\mu_j}$ . In the same way,  $\tilde{\Phi}_{h,p,jk}$  is used to denote the value of  $\tilde{\Phi}(\mathbf{x}, \mu_j, U_k)$  at grid point  $(r_h, \Theta_p)$  as the second gyroaverage with respect to the magnetic moment  $\mu_j$ . It is also approximated by the following right Riemann sum:

$$\tilde{\Phi}_{h,p,jk} = \frac{1}{N_\theta} \sum_{l=0}^{N_\theta-1} \tilde{\Phi} \left( r_h \cos \Theta_p - \rho_j \cos \left( \frac{2\pi l}{N_\theta} \right), r_h \sin \Theta_p - \rho_j \sin \left( \frac{2\pi l}{N_\theta} \right), \mu_j, U_k \right), \quad (33)$$

where

$$\tilde{\Phi}(\mathbf{x}, \mu, U) \equiv (\Phi \mathcal{F}_s)(\mathbf{x}, \mu, U).$$

We should pay attention to the respective symbols + and − in Equations (32) and (33). The cubic splines interpolation is used to calculate  $\phi$  at the sampling point on the Larmor circle. The radial projection on the boundaries for the points outside the domain is used and we consider  $2\pi$ -periodic condition in  $\Theta$ .

The specific procedures to obtain the matrix form of DGT is given by:

1. Construction of the matrix  $A^{spl} \in \mathcal{M}_{(N_r+1) \times (N_\theta), (N_r+1) \times N_\theta}$  such that  $S = A^{spl} \phi$  is the vector of splines coefficients. Alternatively,  $\phi$  can be written as  $\phi = (A^{spl})^{-1} S$ . The matrix  $A^{spl}$  is independent of the Larmor radius or the magnetic moment  $\mu_j$ , while depending on the equilibrium quantities.
2. For a Larmor radius  $\rho_j = \sqrt{2\mu_j}$  contributed by the magnetic moment  $\mu_j$ , constructing the matrix  $A_{l,\rho_j}^{contr} \in \mathcal{M}_{(N_r+1) \times N_\theta, (N_r+1) \times N_\theta}$  which gives the contribution of the gyroaverage of the radius  $\rho_j$  as the function of the splines coefficients. Here,  $l = 1, 2$  denoting the first and the second gyroaverage, respectively.
3. The first gyroaverage is given by  $\Phi_j = G_{1,\rho_j} \phi$ , with  $G_{1,\rho_j} \equiv A_{1,\rho_j}^{contr} A^{spl}$ .
4. The second gyroaverage is written as the matrix form

$$\tilde{\Phi}_{jk} = G_{2,\rho_j} \tilde{\Phi}_{jk}, \quad G_{2,\rho_j} \equiv A_{2,\rho_j}^{contr} A^{spl}.$$

To obtain the matrix form between  $\tilde{\Phi}_{jk}$  and  $\phi$ , the vector product  $\mathcal{F}_{sjk} \cdot \Phi_j$  is written as the product between a diagonal matrix denoted as  $\mathbb{F}_{sjk}$  and  $\Phi_j$ . Eventually,  $\tilde{\Phi}_{jk}$  can be written as the matrix form

$$\tilde{\Phi}_{jk} = G_{2,\rho_j} \mathbb{F}_{sjk} G_{1,\rho_j} \phi. \quad (34)$$

where

$$\mathbb{F}_{sjk} = \begin{bmatrix} \mathbb{F}_{s0jk} & 0 & \cdots & 0 \\ 0 & \mathbb{F}_{s0jk} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbb{F}_{s0jk} \end{bmatrix} \in \mathcal{M}_{(N_r+1) \times N_\theta, (N_r+1) \times N_\theta}(\mathbb{R}) \quad (35a)$$

$$\mathbb{F}_{s0jk} = \begin{bmatrix} \mathcal{F}_{0jk} & 0 & \cdots & 0 \\ 0 & \mathcal{F}_{1jk} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathcal{F}_{N_rjk} \end{bmatrix} \in \mathcal{M}_{N_r \times N_r}(\mathbb{R}) \quad (35b)$$

Please note that the elements of matrix  $G_{2,\rho_j} \mathbb{F}_{sjk} G_{1,\rho_j}$  only depends on the equilibrium quantities. It can be computed once for all at the beginning of the simulation.

5. Then, the integral of  $\mu$  contained by the DGT can be formulated as the discrete sum

$$\tilde{\Phi} = \sum_{k=1}^{N_U} \sum_{j=1}^{N_\mu} \tilde{\Phi}_{jk} B \delta \mu_j \delta U_k$$

Since  $\mathbb{F}_{sjk}$  is a diagonal matrix with only the main diagonal elements not equaling zero, the following arrangements of the product order in Equation (34) equal

$$(G_{2,\rho_j} \mathbb{F}_{sjk}) G_{1,\rho_j} = G_{2,\rho_j} (\mathbb{F}_{sjk} G_{1,\rho_j}).$$

The product order on the left is chosen in this paper. Then, the product  $A^{spl} \mathbb{F}_{sjk}$  is further denoted as  $\mathcal{A}_{jk}^{spl}$  in terms of the following expression

$$\mathcal{A}_{jk}^{spl} = \begin{bmatrix} \mathcal{A}_{0jk}^{spl} & \mathcal{A}_{1jk}^{spl} & \cdots & \mathcal{A}_{(N_\Theta-1)jk}^{spl} \\ \mathcal{A}_{(N_\Theta-1)jk}^{spl} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathcal{A}_{1jk}^{spl} \\ \mathcal{A}_{1jk}^{spl} & \cdots & \mathcal{A}_{(N_\Theta-1)jk}^{spl} & \mathcal{A}_{0jk}^{spl} \end{bmatrix}, \quad (36)$$

where

$$\mathcal{A}_{hjk}^{spl} = A_h^{spl} \mathbb{F}_{s0jk}, h \in \{0, 1, \dots, N_\Theta - 1\}.$$

Then,  $\tilde{\Phi}_{jk}$  in Equation (34) can be written as

$$\tilde{\Phi}_{jk} = A_{2,\rho_j}^{contr} \mathcal{A}_{jk}^{spl} A_{1,\rho_j}^{contr} A^{spl} \phi. \quad (37)$$

#### 4.3. The Fast Algorithm to Solve QNE

Due to the circulant block structure of  $A^{spl}$ ,  $A_{l,\rho_j}^{contr}$  and  $\mathcal{A}_{jk}^{spl}$ , they can be transformed as block diagonal matrixes in Fourier basis [31]. The three matrixes are diagonalizable in the Fourier basis as

$$\begin{aligned} A^{spl} &= U_n D^{spl} U_n^*, \\ A_{l,\rho_j}^{contr} &= U_n D_{l,\rho_j}^{contr} U_n^*, \\ \mathcal{A}_{jk}^{spl} &= U_n \mathcal{D}_{jk}^{spl} U_n^*, \end{aligned}$$

where  $*$  means the complex conjugate and the other matrixes are defined as follows

$$D^{spl} = \begin{pmatrix} D_0^{spl} & & \\ & \ddots & \\ & & D_{N_\Theta-1}^{spl} \end{pmatrix}, \quad (38a)$$

$$D_{l,\rho_j}^{contr} = \begin{pmatrix} D_{l,\rho_j,0}^{contr} & & \\ & \ddots & \\ & & D_{l,\rho_j,N_\Theta-1}^{contr} \end{pmatrix}, \quad (38b)$$

$$\mathcal{D}_{jk}^{spl} = \begin{pmatrix} \mathcal{D}_{jk,0}^{spl} & & \\ & \ddots & \\ & & \mathcal{D}_{jk,N_\Theta-1}^{spl} \end{pmatrix}. \quad (38c)$$

$$D_m^{spl} = \sum_{h=0}^{N_\Theta-1} A_h^{spl} e^{-\frac{2i\pi h m}{N_\Theta}}, \quad (39a)$$

$$D_{l,\rho_j,m}^{contr} = \sum_{h=0}^{N_\Theta-1} A_{l,\rho_j,h}^{contr} e^{-\frac{2i\pi h m}{N_\Theta}}, \quad (39b)$$

$$\mathcal{D}_{jk,m}^{spl} = \sum_{h=0}^{N_\Theta-1} \mathcal{A}_{hjk}^{spl} e^{-\frac{2i\pi h m}{N_\Theta}}, \quad (39c)$$

$$U_n = \begin{pmatrix} U_{n,0,0} & \cdots & U_{n,0,N_\Theta-1} \\ \vdots & \ddots & \vdots \\ U_{n,N_\Theta-1,0} & \cdots & U_{n,N_\Theta-1,N_\Theta-1} \end{pmatrix}, \quad U_{n,h,l} = \frac{1}{\sqrt{N_\Theta}} e^{\frac{2i\pi h l}{N_\Theta}} I_n, \quad (40)$$

where  $I_n$  is the identity matrix of size  $n \times n$ . Then,  $\tilde{\Phi}_{jk}$  in Equation (37) can be written in Fourier basis as

$$\tilde{\Phi}_{jk} = U_n D_{2,\rho_j}^{contr} \mathcal{D}_{jk}^{spl} D_{1,\rho_j}^{contr} D^{spl} U_n^* \phi_F, \quad (41)$$

where  $U_n U_n^* = I_{n \times n}$  is used and  $\phi_F$  is obtained by implementing Fourier transform on the polar elements of the vector  $\phi$  and needs to be solved.

As explained before, the value of  $\phi(\mathbf{x})$  and  $n_1(\mathbf{x})$  on the grid points can be assembled to be the vectors  $\phi$  and  $n_1$ . Then, the first term and third term of Equation (30) can be written in a matrix form with only the main diagonal elements of the coefficient matrix unequal to zero.

By writing  $\tilde{\Phi}_{jk}$  through the products between the matrixes of diagonal block structure based on the Fourier transform, we can design the following fast algorithm to solve QNE regarding the periodic boundary condition in the polar dimension.

1. Fourier transformation of the vector  $n_1$  on the right side of QNE to get the vector  $n_{1F}$  on the Fourier basis.
2. Obtain  $D_m^{spl}$ ,  $D_{l,\rho_j,m}^{contr}$  and  $\mathcal{D}_{jk,m}^{spl}$  by implementing Equation (39) and get the product  $D_{2,\rho_j}^{contr} \mathcal{D}_{jk}^{spl} D_{1,\rho_j}^{contr} D^{spl}$ .
3. Use the subroutine of Lapack and Blas to obtain the inverse matrix  $[(\frac{n_{g0}}{T_i} + \frac{n_{g0}}{T_e}) - D_{2,\rho_j}^{contr} \mathcal{D}_{jk}^{spl} D_{1,\rho_j}^{contr} D^{spl}]^{-1}$ , where  $(\frac{n_{g0}}{T_i} + \frac{n_{g0}}{T_e})$  is of the same structure with  $\mathbb{F}_{sjk}$  in Equation (35a).
4. Compute the vector  $[(\frac{n_{g0}}{T_i} + \frac{n_{g0}}{T_e}) - D_{2,\rho_j}^{contr} \mathcal{D}_{jk}^{spl} D_{1,\rho_j}^{contr} D^{spl}]^{-1} n_{1F}$ .
5. Implementing the inverse Fourier transform on the polar elements of the vector  $[(\frac{n_{g0}}{T_i} + \frac{n_{g0}}{T_e}) - D_{2,\rho_j}^{contr} \mathcal{D}_{jk}^{spl} D_{1,\rho_j}^{contr} D^{spl}]^{-1} n_{1F}$  to obtain  $\phi$ .

The application of polar mesh and FFT enables more quickly computations and provides a base for a future work in more complex geometry.

## 5. Benchmark of the Interpolation Algorithm for the Single $\mu$ Case

To benchmark the interpolation algorithm, we integrate the absolute value of the error of the interpolation solver deviating from the analytical solution over the simulated domain, specifically,

$$\text{abs error} = \sum_{h,p} |(\text{interp})_{hp} - (\text{anal})_{hp}| \delta x_1 \delta x_2, \quad (42)$$

where  $(\text{interp})_{hp}$  and  $(\text{anal})_{hp}$  denote the value computed by the interpolation solver and by the analytical solution at the mesh node  $(h, p)$ , respectively.  $\delta x_1$  and  $\delta x_2$  are the uniform steps in  $x_1$  and  $x_2$  dimensions. For some cases, we also implement the same scheme to the 2nd-order solver.

### 5.1. 1st Example: Double-Periodic Boundary Condition in Cartesian Coordinate

In this example, the periodic boundary condition is used in both  $x_1$  and  $x_2$  dimensions. According to Equations (A12) and (A13), for the function  $f(\mathbf{x}) = \cos(nx_1 + mx_2)$ , the exact solution of  $\mathcal{J}_1(\mu)f(\mathbf{x})$  and  $\mathcal{J}_2(\mu)\mathcal{J}_1(\mu)f(\mathbf{x})$  are

$$\begin{aligned}\mathcal{J}_1(\mu) \cos(nx_1 + mx_2) &= J_0(\rho(\mu)\sqrt{n^2 + m^2}) \cos(nx_1 + mx_2), \\ \mathcal{J}_2(\mu)\mathcal{J}_1(\mu) \cos(nx_1 + mx_2) &= J_0^2(\rho(\mu)\sqrt{n^2 + m^2}) \cos(nx_1 + mx_2).\end{aligned}$$

Two cases are computed. For the first one,  $n = 0, m = 2$  are used, and  $\mu = 0.02, N_\theta = 100$ , while for the second one,  $n = 5, m = 5$  are chosen. For both cases, we compute the domain  $(0, 2\pi) \times (0, 2\pi)$  which are divided into three meshes of (16, 16), (32, 32), (64, 64) cells, respectively. The results are shown in Tables 1 and 2. It is found that compared with the 2nd-order solver, the interpolation algorithm has better accuracy and its results present the converging rate of the fourth-order accuracy, which is provided by the cubic spline interpolation.

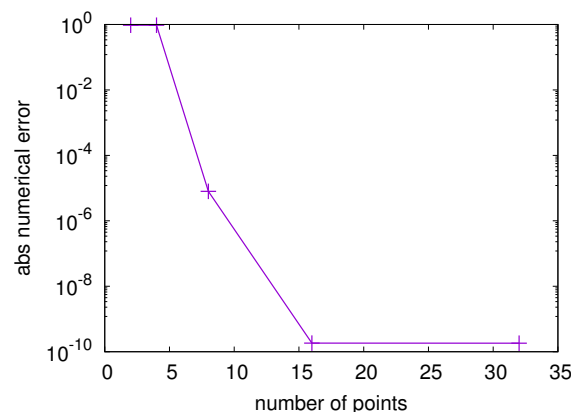
**Table 1.** Double-periodic, Cartesian.  $m = 2, n = 0, \mu = 0.02, N_\theta = 100$ .

| Mesh                     | (16,16)                 | (32,32)                 | (64,64)                 |
|--------------------------|-------------------------|-------------------------|-------------------------|
| interpo, $\mathcal{J}_1$ | $1.7084 \times 10^{-2}$ | $6.4524 \times 10^{-4}$ | $4.1522 \times 10^{-5}$ |
| 2nd, $\mathcal{J}_1$     | 0.8738                  | 0.9635                  | 0.9872                  |
| interpo, $\mathcal{J}_2$ | $3.2803 \times 10^{-2}$ | $1.2393 \times 10^{-3}$ | $7.9757 \times 10^{-5}$ |
| 2nd, $\mathcal{J}_2$     | 1.7102                  | 1.8882                  | 1.9352                  |

**Table 2.** Double-periodic, Cartesian.  $m = 5, n = 5, \mu = 0.02, N_\theta = 100$ .

| Mesh                     | (16,16)  | (32,32)                 | (64,64)                 |
|--------------------------|----------|-------------------------|-------------------------|
| interpo, $\mathcal{J}_1$ | 1.3114   | $4.0744 \times 10^{-2}$ | $2.08 \times 10^{-3}$   |
| 2nd, $\mathcal{J}_1$     | 10.2513  | 10.8218                 | 11.0119                 |
| interpo, $\mathcal{J}_2$ | 1.3972   | $4.5496 \times 10^{-2}$ | $2.3258 \times 10^{-3}$ |
| 2nd, $\mathcal{J}_2$     | 115.6807 | 16.7745                 | 17.1429                 |

For this double-periodic boundary condition, we also check the convergence of the abs error along with the number of interpolating points on the Larmor circle. We use  $m = 2, n = 2$ , the mesh of the rectangular domain is of  $32 \times 32$  cells, and the magnetic moment equals 0.5. The results are presented in Figure 2, which shows that the convergence of the numerical abs error versus the number of the interpolating points on the Larmor circle.



**Figure 2.** The numerical abs error versus the number of sampling points on the Larmor circle.

### 5.2. 2nd Example: Natural + Periodic Boundary Condition in Cartesian Coordinate

For this case, the periodic boundary condition is used in  $x_2$  dimension, while the natural boundary condition is used in  $x_1$  dimension. The function under gyroaverage is  $f(\mathbf{x}) = x_2$ . Its first gyroaverage and double gyroaverage are

$$\mathcal{J}_1(f(\mathbf{x})) = x_2, \mathcal{J}_2\mathcal{J}_1(f(\mathbf{x})) = x_2.$$

For this case,  $\mu = 0.05$ . The simulated domain  $(0, 4) \times (0, 2\pi)$  are divided into two meshes including  $32 \times 32$  and  $64 \times 64$  cells, respectively. The integrals of the absolute value of the error over the simulated domain for the first and double gyroaverage are presented in Table 3. To obtain this, we ignored the boundary rows in the  $x_1$  dimension, for which the natural boundary condition is used. The fact that the abs error of the 2nd-order solver is zero is due to the fact that the first gyroaverage and the double one of  $f(\mathbf{x})$  depends on the gradient of  $f(\mathbf{x})$  only through  $\nabla_{\perp}^2 f(\mathbf{x})$ , as Equation (A3) shows.  $\nabla_{\perp}^2 f(\mathbf{x})$  in Cartesian coordinates is written as  $(\partial_{x_1}^2 + \partial_{x_2}^2)f(\mathbf{x})$ . The second order derivative is approximated by the centered finite difference as  $\partial_{x_1}^2 f(\mathbf{x}) = \frac{f_{j+1,k} - 2f_{j,k} + f_{j-1,k}}{\delta x_1^2}$  with the 2nd-order precision, where  $f_{j+1,k} \equiv f(x_{1,j+1}, x_{2,k})$ . It is obvious that the finite difference of  $\partial_{x_1}^2$  over  $f(\mathbf{x})$  being  $x_2$  equals zero, so does  $\partial_{x_2}^2$  over  $f(\mathbf{x})$  except at the boundary. The abs error of the 2nd-order solver equaling zero is obtained when the boundary row are ignored.

**Table 3.** Natural + periodic, Cartesian.  $f(\mathbf{x}) = x_2, \mu = 0.05, N_{\theta} = 100$ .

| Mesh                     | (32,32)                 | (64,64)                 |
|--------------------------|-------------------------|-------------------------|
| interpo, $\mathcal{J}_1$ | $1.0297 \times 10^{-3}$ | $1.7507 \times 10^{-2}$ |
| 2nd, $\mathcal{J}_1$     | 0.0                     | 0.0                     |
| interpo, $\mathcal{J}_2$ | $2.3478 \times 10^{-2}$ | $8.3417 \times 10^{-2}$ |
| 2nd, $\mathcal{J}_2$     | 0.0                     | 0.0                     |

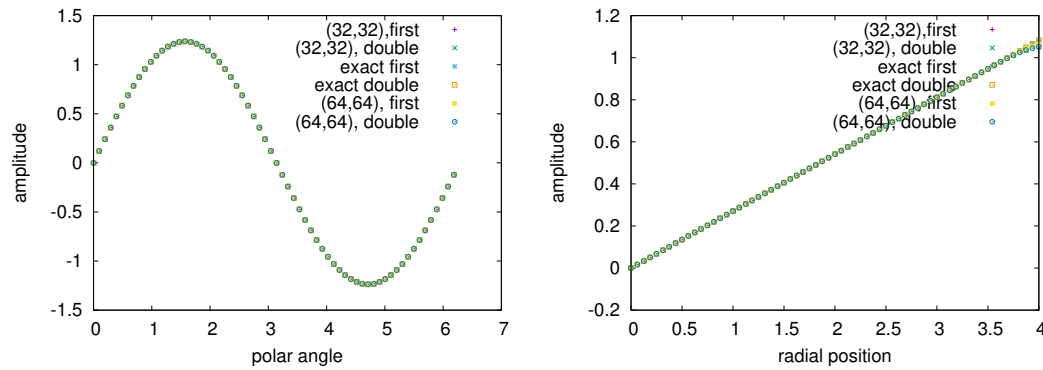
### 5.3. 3rd Example: Natural + Periodic Boundary Condition in the Polar Coordinate Frame

For this case, the periodic boundary condition is used in  $\Theta$  dimension, while the natural boundary condition used in  $r$  dimension. We also use the test function  $f(\mathbf{x}) = x_2$ , and the domain is also chosen as  $(0, 4) \times (0, 2\pi)$ . The integral of the absolute value of the error is presented in Table 4. The boundary rows in  $r$  dimension are ignored to get these results. The exact profile of the first and double gyroaverage of the test function at the radial position  $7/4$  along the polar dimension and the corresponding numerical solutions computed in the two meshes are plotted in the left figure of Figure 3. The exact profile of the first and double gyroaverage of the test function profile at the polar angle  $7\pi/8$  along the radial dimension and the corresponding numerical solutions computed in the two meshes are plotted in the right figure of Figure 3. Both figures show that the results computed by the interpolation algorithm fit the exact value very well. However, we can still notice a small error at the outer boundary of  $x_1$  dimension, where the natural boundary condition is used.

**Table 4.** Natural + periodic, polar.  $f(\mathbf{x}) = x_2, \mu = 0.02, N_{\theta} = 100$ .

| Mesh                     | (32,32)                 | (64,64)                 |
|--------------------------|-------------------------|-------------------------|
| interpo, $\mathcal{J}_1$ | $8.0966 \times 10^{-4}$ | $1.0698 \times 10^{-5}$ |
| interpo, $\mathcal{J}_2$ | $1.9712 \times 10^{-2}$ | $2.9669 \times 10^{-3}$ |





**Figure 3.** The left figure shows the exact value of the 1st and double gyroaverage of the test function and the value computed in the two meshes of the polar profile at radial position  $7/4$ . The right figure shows the radial profile at the polar angle  $7\pi/8$ .

#### 5.4. Benchmark the QNE Solver in the Cartesian Coordinate Frame and in the Polar Coordinate Frame

The previous examples benchmark the interpolation algorithm to compute the gyroaverage operation. This subsection is dedicated to benchmark the QNE solver, which comprises solving the inverse matrix of  $A^{spl}$  and  $A_{l,\rho_j}^{contri}$  by FFT based on the algorithm given in Subsec.(4.2), and the interpolation algorithm to obtain the gyroaverage.

For the convenience to find the exact solution of QNE, we simplify QNE in Equation (22) by choosing  $T_i = T_e = 0.5$ ,  $n_0 = n_{g0} = 1.0$ . We also solve the single  $\mu$  case first, so that QNE is simplified as

$$4\phi(\mathbf{x}) - \mathcal{J}_2(\mu)\mathcal{J}_1(\mu)\phi(\mathbf{x}) = n_1(\mathbf{x}). \quad (43)$$

We first benchmark the QNE solver in Cartesian coordinate frame with double-periodic boundary condition. By choosing

$$n_1(\mathbf{x}) = \cos(nx_1 + mx_2),$$

the exact solution of Equation (43) is

$$\phi(\mathbf{x}) = \frac{\cos(nx_1 + mx_2)}{4 - J_0^2(\rho(\mu)\sqrt{n^2 + m^2})}.$$

We choose  $n = 3$ ,  $m = 3$ ,  $\mu = 1.0$  and  $\mu = 0$ . The simulated domain  $(0, 4) \times (0, 2\pi)$  are divided into three meshes including  $32 \times 32$ ,  $64 \times 64$  and  $128 \times 128$  cells, respectively. The integrals of the absolute value of the error for the three kinds of mesh are presented in Table 5, which indicates that the QNE solver could provide a good solution. For the  $\mu = 0.5$  case, we also plot in Figure 4 the exact solution and the results computed by the QNE solver with node number in  $x_2$  dimension equaling 20. The two solutions in Figure 4 fit each other well.

To benchmark the QNE solver in the polar coordinate frame, the function  $n_1$  in Equation (43) is chosen as

$$n_1(\mathbf{x}) = x_2,$$

which leads to the exact solution of  $\phi(\mathbf{x})$  through Equation (43)

$$\phi(\mathbf{x}) = x_2.$$

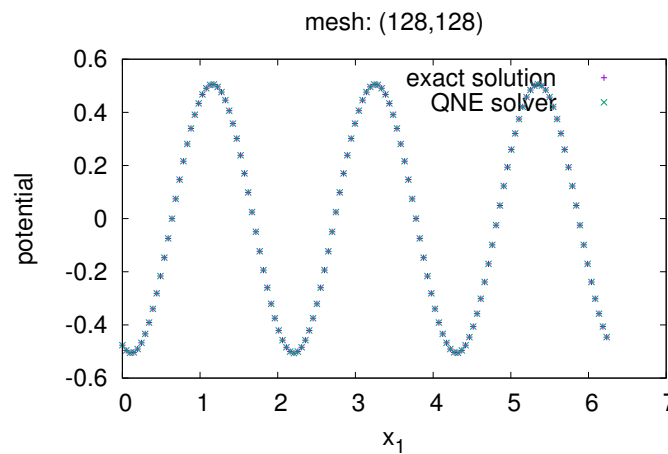
The simulated domain  $(0, 4) \times (0, 2\pi)$  is also divided into meshes of  $32 \times 32$  and  $64 \times 64$  cells, respectively. We computed two cases of  $\mu = 0.02$  and  $\mu = 0$ . The integral of the absolute error over the simulated domain for the two  $\mu$  cases are given in Table 6. The fourth-order converging rate does not appear anymore, due to the error at the boundary area where the natural boundary condition is used. The error has two sources. One is that the natural boundary condition is used. The other is that

around boundary, due to the finite value of the Larmor radius, there are sampling points located out of the simulated domain and we project these points to the corresponding points at the boundary.

We still plot the exact solution and the numerical solutions of the polar profile at the radial position  $7/4$  and the radial profile at the polar angle  $7\pi/8$  in Figure 5, which presents good match between the two solutions.

**Table 5.** Double-periodic, Cartesian.  $n = 3, m = 3, N_\theta = 100$ .

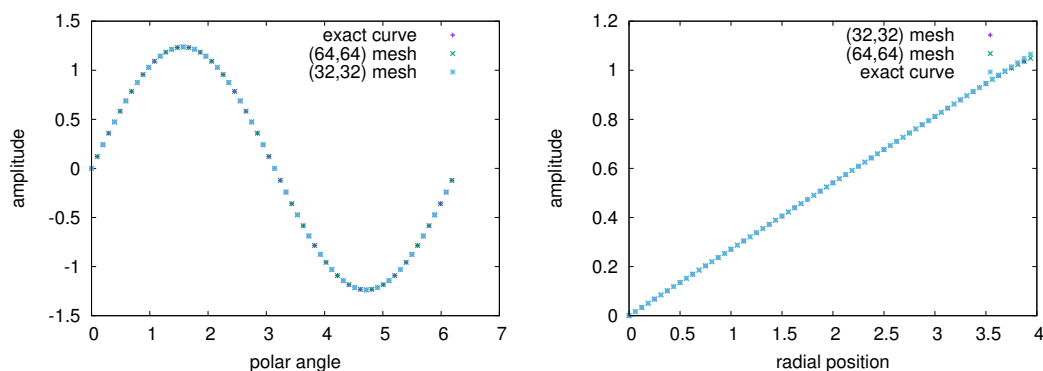
| Mesh        | (32,32)                  | (64,64)                  | (128,128)                |
|-------------|--------------------------|--------------------------|--------------------------|
| $\mu = 1.0$ | $1.5204 \times 10^{-4}$  | $6.5066 \times 10^{-6}$  | $2.9072 \times 10^{-7}$  |
| $\mu = 0.0$ | $1.2393 \times 10^{-13}$ | $1.4944 \times 10^{-13}$ | $1.3548 \times 10^{-13}$ |



**Figure 4.** The exact solution and the numerical solution solved by the QNE solver at node 20 in the  $x_2$  dimension.

**Table 6.** Natural + periodic, polar.  $f(\mathbf{x}) = x_2, N_\theta = 100$ .

| Mesh         | (32,32)                  | (64,64)                  |
|--------------|--------------------------|--------------------------|
| $\mu = 0.02$ | 0.1186                   | 0.1339                   |
| $\mu = 0.0$  | $2.6356 \times 10^{-13}$ | $1.4944 \times 10^{-13}$ |



**Figure 5.** The left figure shows the exact solution and the numerical solutions of the polar profile at the radial position  $7/4$ . The right figure shows the radial profile at polar angle  $7\pi/8$ .

## 6. Benchmark of the Interpolation Algorithm for the Multiple $\mu$ Case

This benchmark is done in the polar coordinate frame. The test function is chosen as

$$f(\mathbf{x}) = \cos(nx_1 + mx_2).$$

We consider the gyroaverage associated with  $\mu$  obeying the distribution  $e^{-\mu/a}$  where  $a \in (0.5, 2.2)$  is chosen.

The integral of the distribution over  $\mu$  is written as

$$\mathcal{I}_0(\mu_{\max}, n, m) = \int_0^{\mu_{\max}} e^{-\mu/a} d\mu$$

and the exact solution of the gyroaverage of the test function  $f(\mathbf{x})$  are related to the following two functions

$$\mathcal{I}_1(\mu_{\max}, n, m) = \int_0^{\mu_{\max}} J_0(\rho(\mu) \sqrt{n^2 + m^2}) e^{-\mu/a} d\mu$$

and

$$\mathcal{I}_2(\mu_{\max}, n, m) = \int_0^{\mu_{\max}} J_0^2(\rho(\mu) \sqrt{n^2 + m^2}) e^{-\mu/a} d\mu.$$

The exact solution of the first gyroaverage is

$$\int_0^{\mu_{\max}} \mathcal{J}_1(\mu) f(\mathbf{x}) e^{-\mu/a} d\mu = \mathcal{I}_1(\mu_{\max}, n, m) f(\mathbf{x}),$$

and the exact solution of the double gyroaverage is

$$\int_0^{\mu_{\max}} \mathcal{J}_2(\mu) \mathcal{J}_1(\mu) f(\mathbf{x}) e^{-\mu/a} d\mu = \mathcal{I}_2(\mu_{\max}, n, m) f(\mathbf{x}),$$

which can be written into a discretized formula

$$\sum_{i=1}^{N_\mu} \mathcal{J}_2(\mu_i) \mathcal{J}_1(\mu_i) f(\mathbf{x}) e^{-\mu_i/a} \delta\mu_i.$$

### 6.1. The Euler–Maclaurin-Based Quadrature Integration Algorithm

$\tilde{\Phi}$  is given in Equation (23) and  $F$  is proportional to  $\exp(-\frac{\mu B}{T_i(r)})$ . To obtain the integral over  $\mu$ , the conventional way is to redefine a new magnetic moment by  $\mu_{new} = \mu / T_i(r)$ . Then, the exponential factor  $\exp(-\frac{\mu B}{T_i(r)})$  becomes  $\exp(-\mu_{new})$  and the Gauss-Laguerre quadrature is implemented to obtain the discretized sum to approximate the continuous integral. However, in the new term  $\tilde{\Phi}$ , the denominator of  $T_i(r)$  in  $\exp(-\frac{\mu B}{T_i(r)})$  together with  $n_0(r)$  determines the radial density profile of the ions which possess the magnetic moment  $\mu$ . As  $T_i(r)$  could possess the steep gradient in the radial dimension, its gyroaverage  $\mathcal{J}(\mu) T_i(\mathbf{x})$  cannot be ignored, resulting that  $T_i(r)$  can not be eliminated from the denominator in the conventional way. Moreover, we cannot use Gauss-Laguerre quadrature method to provide the same group of  $\mu$  nodes and the associated weights along the radial dimension for different  $T_i(r)$  in the radial domain. Because the value of  $T_i(r)$  is within a domain, which is chosen as (0.5, 2.2) in this paper, and the gradient of  $T_i(r)$  after the gyroaverage could also change, the roots and weights provided by Gauss-Laguerre quadrature for the integral  $\int_0^{\mu_{\max}} e^{-\mu} d\mu$  can not fit the integral  $\int_0^{\mu_{\max}} e^{-\mu/a} d\mu$  anymore for all  $a$ s belonging to the domain (0.5, 2.2).

We therefore develop a quadrature integrating scheme based on the following Euler–Maclaurin quadrature formula

$$\int_b^c f(x) dx = \frac{h}{2} (f(b) + f(c)) + \sum_{k=1}^{n-1} h f(b + kh) - \sum_{r=1}^M \frac{h^{2r} B_{2r}}{(2r)!} (f^{2r-1}(c) - f^{2r-1}(b)) + R_p. \quad (44)$$

where  $h = (c - b)/n$ ,  $n - 1$  is the number of the cells of the mesh,  $B_{2r}$  is the  $2r$ -th Bernoulli number and  $R_p$  is the residual quantity. The order of the precision of the right side is beyond  $O(h^{2M})$ . For the distribution function proportional to  $\exp(-\mu/a)$ ,  $R_p$  and the term depending on  $f^{2r-1}(c)$  can be ignored for large enough  $c$ .  $B_{2r}$  can be solved by the recurrence relation

$$B_p = -\frac{1}{p+1} \sum_{k=0}^{p-1} \binom{p+1}{k} B_k.$$

The derivative  $f^{2r-1}(b)$  is replaced by the finite difference scheme. For a function  $F(x)$ , the finite difference scheme of the  $d$ th order derivative with the  $p$ th order precision can be formally written as the following summation of the function value on the grid points

$$\frac{h^d}{d!} F^{(d)}(x) = \sum_{i=i_{\min}}^{i_{\max}} G_{dpi} F(x+ih) + O(h^{d+p}), \quad (45)$$

where  $G_{dpi}$  is the coefficient of the finite difference with the  $p$ th order precision for the  $d$ th order derivative, and the step length of the finite difference is the same with that used in Equation (44). For the forward finite difference scheme of the  $d$ th order derivative with  $p$ th order precision,  $i_{\min} = 0$  and  $i_{\max} = d + p - 1$  are derived, while for the centered finite difference scheme, we have  $-i_{\min} = i_{\max} = (d + p - 1)/2$ . By truncating the Taylor expanding of  $F(x + ih)$  at the order  $O((ih)^{d+p-1})$ , Equation (45) can be rewritten as

$$\begin{aligned} F^{(d)}(x) &= \frac{d!}{h^d} \sum_{n=0}^{d+p-1} \left( \sum_{i=i_{\min}}^{i_{\max}} i^n G_{dpi} \right) \frac{h^n}{n!} F^{(n)}(x) + O(h^p) \\ &\approx \frac{d!}{h^d} \sum_{n=0}^{d+p-1} \left( \sum_{i=i_{\min}}^{i_{\max}} i^n G_{dpi} \right) \frac{h^n}{n!} F^{(n)}(x). \end{aligned} \quad (46)$$

The equaling of both sides of Equation (46) results in the following identities

$$\sum_{i=i_{\min}}^{i_{\max}} i^n G_{dpi} = \begin{cases} 0, 0 \leq n \leq d + p - 1 \text{ and } n \neq d \\ 1, n = d \end{cases},$$

through which the coefficient  $G_{dpi}$ s can be solved. By substituting Equation (46) into the Euler–Maclaurin formula, we obtain its discretized version

$$\int_b^c f(x) dx = \frac{h}{2} (f(b) + f(c)) + \sum_{k=1}^{n-1} h f(b + kh) + h \sum_{r=1}^M \frac{B_{2r}}{2r} \sum_{i=i_{\min}}^{i_{\max}} G_{(2r-1)pi} f(b + ih). \quad (47)$$

For the forward finite difference scheme, Equation (47) can be rearranged into

$$\begin{aligned} \int_b^c f(x) dx &= \sum_{i=1}^{2M+p-2} \left[ h + h \sum_{\substack{1 \leq r \leq M, \\ i \leq 2r+p-2}} \frac{B_{2r}}{2r} G_{(2r-1)pi} \right] f(b + ih) \\ &+ \left[ \frac{h}{2} + h \sum_{r=1}^M \frac{B_{2r}}{2r} G_{(2r-1)p0} \right] f(b) + \sum_{i=2M+p-1}^{n-1} h f(b + ih) + \frac{h}{2} f(c), \end{aligned} \quad (48)$$

where we select the condition satisfying  $2M + p - 1 \leq n - 1$ . The weight coefficients of  $f(b + ih)$  in Equation (48) for all  $i \in \{0, 1, 2, \dots, n - 1\}$  are computed and stored at the beginning of the simulation for the following quadrature integration.

For the central difference scheme, Equation (47) can be rearranged in the same way into

$$\int_b^c f(x)dx = \sum_{i=1}^{(2M+p-2)/2} \left[ hf(b+ih) + h \sum_{\substack{1 \leq r \leq M, \\ i \leq (2r+p-2)/2}} \frac{B_{2r}}{2r} \left( G_{(2r-1)p} f(b+ih) + G_{(2r-1)p(-i)} f(b-ih) \right) \right] \\ + \left[ \frac{h}{2} + h \sum_{r=1}^M \frac{B_{2r}}{2r} G_{(2r-1)p0} \right] f(b) + \sum_{i=(2M+p)/2}^{n-1} hf(b+ih) + \frac{h}{2} f(c), \quad (49)$$

The value of  $f(b-ih)$  is obtained by considering the even or odd property of  $f(x-b)$  relative to the point  $x=b$ . The value of  $b$  equals 0 for the situation we consider, while the integrand is an even function relative to the point  $b=0$ .

One more point to mention is that instead of  $\mu$ , we implement  $v_{\perp}$ , which equals  $\sqrt{2\mu}$ , as the integration argument for the achievement of the high precision. So  $d\mu = v_{\perp} dv_{\perp}$  and the minimal value of  $v_{\perp}$  equals 0. Based on this implementation, we developed the third one named derivative-reduction scheme. By replacing  $\mu$  with  $v_{\perp}$ , the integral  $\int f(2\mu)d\mu$  becomes  $\int f(v_{\perp}^2)v_{\perp}dv_{\perp}$ , with the new integrand being  $g(v_{\perp}) \equiv f(v_{\perp}^2)v_{\perp}$ . At the point  $v_{\perp}=0$ , we use the following formula to reduce the order of the derivative

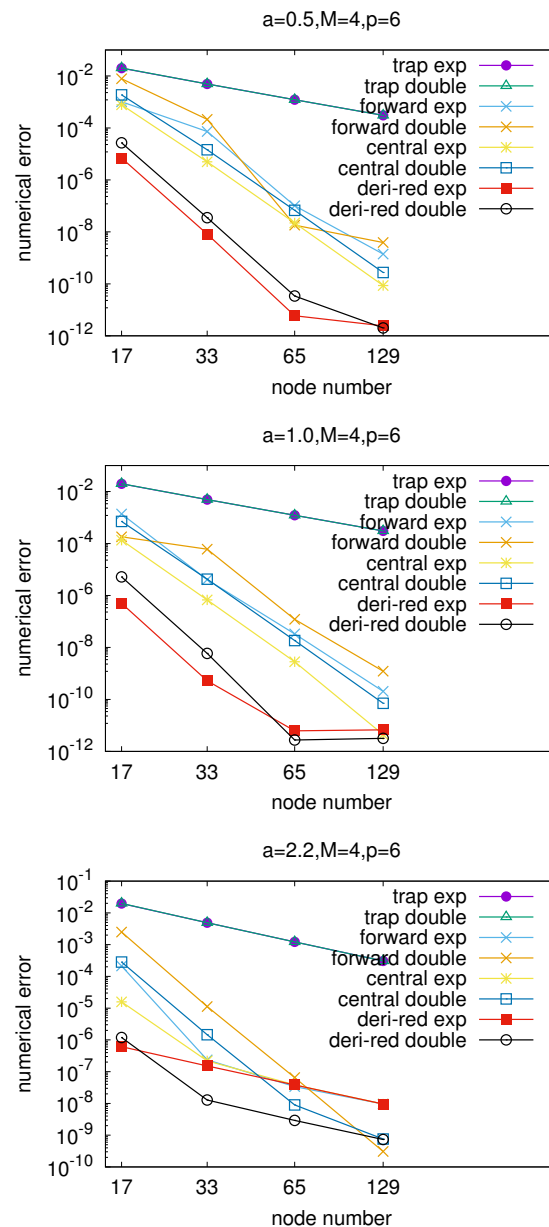
$$g^{2k-1}(0) = (2k-1)f^{2k-2}(0).$$

Then, the third discretized expression of Euler–Maclaurin formula is derived as

$$\int_0^c g(v_{\perp})dv_{\perp} \approx \frac{h}{2}(g(0) + g(c)) + \sum_{k=1}^{n-1} hg(kh) + \sum_{r=1}^M \frac{h^{2r}B_{2r}}{(2r)!} g^{2r-1}(0) \\ = \frac{h}{2}(g(0) + g(c)) + \sum_{k=1}^{n-1} hg(kh) + \sum_{r=1}^M \frac{h^{2r}B_{2r}}{(2r)!} (2r-1)f^{2r-2}(0) \\ = \frac{h}{2}(g(0) + g(c)) + \sum_{k=1}^{n-1} hg(kh) + \frac{h^2B_2}{2}f(0) + h^2 \sum_{r=2}^M \frac{B_{2r}}{2r} \sum_{i=i_{\min}}^{i_{\max}} G_{(2r-2)p} f(0) \\ = \frac{h}{2}cf(c) + \sum_{k=1}^{n-1} h(kh)f(kh) + \frac{h^2B_2}{2}f(0) + h^2 \sum_{r=1}^M \frac{B_{2r+2}}{2r+2} \sum_{i=i_{\min}}^{i_{\max}} G_{(2r)p} f(0) \quad (50) \\ = \sum_{i=1}^{(2M+p-1)/2} \left[ h(ih)f(ih) + h^2 \sum_{\substack{1 \leq r \leq M, \\ i \leq (2r+p-1)/2}} \frac{B_{2r+2}}{2r+2} (G_{(2r)p} + G_{(2r)p(-i)})f(ih) \right] \\ + \frac{h^2B_2}{2}f(0) + h^2 \sum_{r=1}^M \frac{B_{2r+2}}{2r+2} G_{(2r)p0}f(0) + \sum_{i=(2M+p-1)/2+1}^{n-1} h(ih)f(ih) + \frac{h}{2}cf(c),$$

where we have used the even property of the function  $f(v_{\perp}^2)$  at  $v_{\perp}=0$ , so that  $f(-ih) = f(ih)$ .

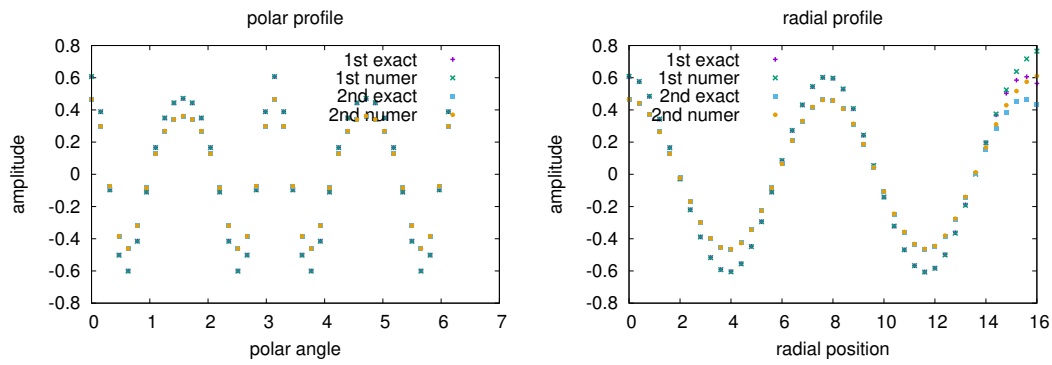
To test the accuracy and the convergent rate of the three schemes based on the Euler–Maclaurin formula, we apply them to the computation of the exponential integral  $\int_0^{30} e^{-\mu/a}d\mu$  and the associated double-gyroaverage integral  $\int_0^{30} \int_0^2 (\sqrt{2\mu})e^{-\mu/a}d\mu$ . In addition, we choose  $M=4, p=6$  for the test purpose and for the following other numerical simulation examples. We also compare its results to those computed by the trapezoidal quadrature algorithm with uniform mesh. Through dividing the domain  $(0, 30)$  into  $10^7$  equal cells, the two integrals are approximated by the Riemann sum. By treating the Riemann sums as the exact value, the absolute value of the numerical error computed by the two algorithms are plotted in Figure 6 for the three cases  $a=0.5, 1.0, 2.2$ . Figure 6 shows that compared with the trapezoidal quadrature algorithm, the three schemes of the Euler–Maclaurin-based algorithm provide a much higher precision order when the nodes number goes from 17 to 129.



**Figure 6.** The three figures show the magnitude of the numerical error of the quadrature integral of  $\int_0^{30} e^{-\mu/a} d\mu$  and  $\int_0^{30} J_0^2(\sqrt{2\mu}) e^{-\mu/a} d\mu$  computed by the three schemes of Euler–Maclaurin-based quadrature integrating algorithm and the trapezoidal algorithm with uniform mesh. We choose  $M = 4$  in the Euler–Maclaurin formula and  $p = 6$  as the precision order of the finite difference. Here, “deri-red” denotes the derivative-reduction scheme, while “trap” denotes the trapezoidal algorithm.

## 6.2. Benchmark of DGT Computed by the Interpolation Algorithm in the Polar Coordinate Frame Assisted by the Forward Scheme of the Euler–Maclaurin-Based Algorithm

This benchmark is done in the polar coordinate frame and the forward scheme of the Euler–Maclaurin-based algorithm is implemented. In this benchmark,  $n = 0, m = 1$  is chosen. The simulated domain of  $(r, \Theta)$  is  $(0, 16) \times (0, 2\pi)$ , which is divided into  $40 \times 40$  meshes.  $\mu$  is in  $(0, 20)$  which is divided into one mesh including 64 nodes. Only the case of  $a = 1$  is computed here. The polar profile at  $r = 6$  and the radial profile at  $\Theta = \frac{7\pi}{8}$  are shown in Figure 7.



**Figure 7.** The left figure shows the exact and numerical polar profile at the radial position  $r = 6$ , while the right figure shows the exact and numerical radial profile at  $\Theta = \frac{7\pi}{8}$ .

### 6.3. Benchmark of the QNE Solver in the Polar Coordinate Frame Assisted by the Forward Scheme of the Euler–Maclaurin–Based Algorithm

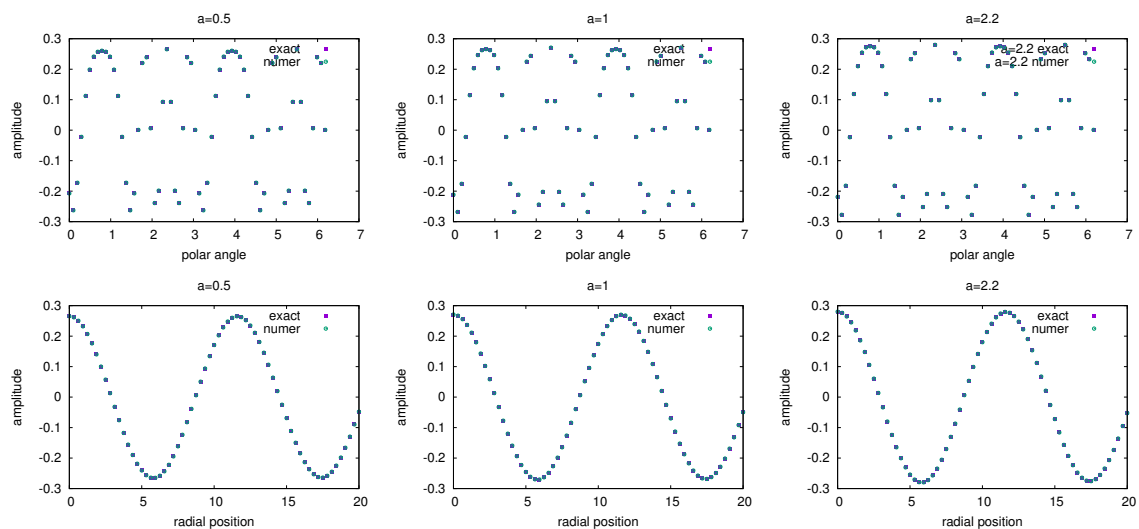
We also benchmark the QNE solver in the polar coordinate frame with  $\mu$  obeying the distribution  $e^{-\mu/a}$ , with which the quasi-neutrality equation in this case is

$$4\phi(\mathbf{x}) - \int_0^{\mu_{\max}} \mathcal{J}_2(\mu) \mathcal{J}_1(\mu) \phi(\mathbf{x}) e^{-\mu/a} d\mu = n_1(\mathbf{x}), \quad (51)$$

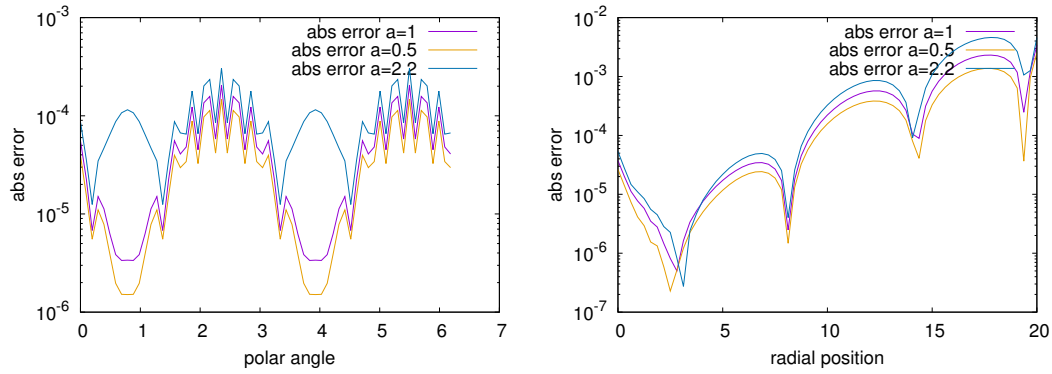
where the same parameters  $T_{i0}, T_{e0}, n_0, n_{g0}$  are used. We choose the test function of  $n_1(\mathbf{x}) = \cos(nx_1 + mx_2)$ . The exact solution of Equation (51) is

$$\phi(\mathbf{x}) = \frac{\cos(nx_1 + mx_2)}{4 - \mathcal{I}_2(\mu_{\max}, n, m)}.$$

In this benchmark, the radial domain is chosen as  $(0, 20)$ . The domain of  $r \times \Theta$  is divided into a mesh of  $64 \times 64$  cells. The weights of the  $\mu$  mesh for  $a = 1, 0.5, 2.2$  are computed by the Euler–Maclaurin-based quadrature algorithm. The magnitude of the numerical error of  $\mathcal{I}_0, \mathcal{I}_2$  and the abs error of the integration of the solution over the domain for the three  $a$  cases are given in Table 7. The computing formula of the abs error is presented by Equation (42). In Figure 8, the radial profile at  $\Theta = \frac{7\pi}{8}$  and the polar profile at  $r = \frac{35}{4}$  for  $a = 0.5, a = 1, a = 2.2$  cases are plotted. The absolute value of the numerical error of the polar profile and the radial profile are plotted in the left and right figure of Figure 9.



**Figure 8.** The radial profile at  $\Theta = \frac{7\pi}{8}$  and the polar profile at  $r = \frac{35}{4}$  for  $a = 0.5, a = 1, a = 2.2$  cases are plotted.



**Figure 9.** The left figure shows the abs error of the polar profile with  $r = \frac{35}{4}$ , while the right one presents the abs error of the radial profile with  $\Theta = \frac{7\pi}{8}$ .

**Table 7.**  $N_\mu = 64, n = 1, m = 1, \mu_{\max} = 20, N_\theta = 100$ .

|                             | $a = 1.0$               | $a = 0.5$               | $a = 2.2$               |
|-----------------------------|-------------------------|-------------------------|-------------------------|
| $\mathcal{I}_0$ , num error | $2.4991 \times 10^{-5}$ | $2.4955 \times 10^{-5}$ | $2.662 \times 10^{-5}$  |
| $\mathcal{I}_2$ , num error | $2.4908 \times 10^{-5}$ | $2.4925 \times 10^{-5}$ | $2.4915 \times 10^{-5}$ |
| solution abs error          | 1.3817                  | 0.653                   | 3.6886                  |

## 7. Application of the QNE Solver in Gyrokinetic Simulation Concerning Short-Scale Perturbation and Steep Equilibrium Profile Assisted by the Forward Scheme of the Euler–Maclaurin-Based algorithm

### 7.1. The Gyrokinetic Model with Constant Cylindrical Magnetic Field Configuration

The cylindrical coordinates frame with the constant theta-pinch magnetic field configuration is used in our simulations. For this configuration adapted in this paper, the equilibrium magnetic field is along the  $z$  direction and its magnitude is constant in the cross surface. The periodic boundary condition is used in the parallel dimension and in the polar dimension, while the natural boundary condition in the radial dimension. The whole simulations are executed on the ATLAS HPC platform of IRMA by introducing our test modules into SeLaLiB code [32], which is a classic semi-Lagrangian library using cubic splines interpolation [33–36]. The predictor-corrector method, the Verlet algorithm for computing the characteristics, and the Strang splitting of the advection of Vlasov equation are also implemented [35,37–40].

The gyrokinetic Vlasov equation in cylindrical coordinate frame is

$$\left[ \partial_t + \left( \frac{1}{r} \partial_\Theta \Phi \partial_r - \frac{1}{r} \partial_r \Phi \partial_\Theta \right) + U \partial_{x_\parallel} + \partial_{x_\parallel} \Phi \partial_U \right] F = 0. \quad (52)$$

The characteristics are

$$\dot{r} = \frac{1}{r} \partial_\Theta \Phi, \quad (53a)$$

$$\dot{\Theta} = -\frac{1}{r} \partial_r \Phi, \quad (53b)$$

$$\dot{x}_\parallel = U, \quad (53c)$$

$$\dot{U} = \partial_{x_\parallel} \Phi \quad (53d)$$

The quasi-neutrality equation is given by Equation (22).



## 7.2. The Parallelization

### 7.2.1. The Algorithm to Reduce the Distribution to the Density

In the discrete version of  $\mu$ , the distribution of ions associated with each  $\mu_j$  with  $j \in \{1, \dots, N_\mu\}$  is denoted as  $F_j(\mathbf{x}, \mu_j, U)$ . Due to the identity  $\frac{d\mu_j}{dt} = 0$ ,  $F_j(\mathbf{x}, \mu_j, U)$  satisfies the Vlasov equation in the cylindrical coordinate system

$$[\partial_t + (\frac{1}{r} \partial_\Theta \Phi(\mathbf{x}, \mu_j) \partial_r - \frac{1}{r} \partial_r \Phi(\mathbf{x}, \mu_j) \partial_\Theta) + U \partial_{x_\parallel} + \partial_{x_\parallel} \Phi(\mathbf{x}, \mu_j) \partial_U] F_j(\mathbf{x}, \mu_j, U) = 0, \quad (54)$$

where  $(r, \Theta, x_\parallel)$  denotes the cylindrical coordinate frame.  $F_j(\mathbf{x}, \mu_j, U)$  can be rewritten as the sum

$$F_j(\mathbf{x}, \mu_j, U) = F_{0j}(\mathbf{x}, \mu_j, U) + F_{1j}(\mathbf{x}, \mu_j, U),$$

with

$$F_{0j}(\mathbf{x}, \mu_j, U) = F_{0j\parallel}(\mathbf{x}, U) F_{0j\perp}(\mathbf{x}, \mu_j) \quad (55)$$

and  $F_{0j\perp}(\mathbf{x}, \mu_j) = \frac{1}{2\pi T_i} \exp(-\frac{\mu_j B(\mathbf{x})}{T_i(\mathbf{x})})$ . In the numerical simulation,  $F_{0j}(\mathbf{x}, \mu_j, U)$  does not evolve. At each time step,  $F_j(\mathbf{x}, \mu_j, U)$  is obtained by solving Equation (54) and  $F_{1j}(\mathbf{x}, \mu_j, U)$  is derived by using  $F_j(\mathbf{x}, \mu_j, U)$  minus  $F_{0j}(\mathbf{x}, \mu_j, U)$ .

The normalized version of Equations (17a) and (17b) to obtain  $n_{sg0}$  and  $n_{sg1}$  can be discretized in the  $\mu$  dimension by obtaining a mesh of  $\mu$  with  $N_\mu$  nodes. The contribution of each  $\mu_j$  component to the density  $n_{sg}$  and  $n_{sg0}$  can be written as

$$\begin{aligned} n_{sgj}(\mathbf{x}) &= B(\mathbf{x}) \delta \mu_j \int F_s(\mathbf{x} - \rho_0(\mathbf{x}, \mu_j), \mu_j, U) dU, \\ n_{sg0j}(\mathbf{x}, t) &= B(\mathbf{x}) \delta \mu_j \int F_{s0}(\mathbf{x} - \rho_0(\mathbf{x}, \mu_j), \mu_j, U) dU. \end{aligned}$$

Then, the  $n_{sg1j}$  can be derived as  $n_{sgj} - n_{sg0j}$ . In addition, the summation of all  $\mu_j$ s leads to the total density. The specific procedures to compute the density on the particle coordinate frame are listed below:

1. The distribution  $F_j(\mathbf{x}, \mu_j, U, t)$  for each  $\mu_j$  at time moment  $t$  is computed by Equation (54).
2. The integration of the distribution of each  $F_j(\mathbf{x}, \mu_j, U, t)$  over  $U$  is first done by the sum  $\mathcal{S}_j(\mathbf{x}, t) \equiv \sum_j F_j(\mathbf{x}, \mu_j, U_k, t) \delta U$ .
3. The gyroaverage of  $\mathcal{S}_j(\mathbf{x}, t)$  denoted as  $\mathcal{J}_2(\mu_j) \mathcal{S}_j(\mathbf{x}, t)$  for each  $\mu_j$  is done by using the coefficient matrix with the result being  $A_{2,\rho_j}^{contr} A^{spl} \mathcal{S}_{jt}$ , where  $\mathcal{S}_{jt}$  is a vector formed by  $\mathcal{S}_j(\mathbf{x}, t)$  on the grids and is given as

$$\mathcal{S}_{jt} := (\mathcal{S}_{j,0,0}, \dots, \mathcal{S}_{j,N_r,0}, \mathcal{S}_{j,0,1}, \dots, \mathcal{S}_{j,N_r,1}, \dots, \mathcal{S}_{j,0,N_\Theta-1}, \dots, \mathcal{S}_{j,N_r,N_\Theta-1})^T$$

The total density on the particle coordinate mesh contributed by the  $\mu_j$  component is written as

$$n_{gjh,p} \equiv (\mathcal{J}_2(\mu_j) \mathcal{S}_j)_{hp}, \quad h \in \{0, \dots, N_r\}, p \in \{0, \dots, N_\Theta - 1\}.$$

4. The equilibrium density in the particle coordinate frame obtained from the gyroaverage of the equilibrium distribution  $F_{0j}(\mathbf{x}, \mu_j, U)$  for the  $\mu_j$  component is done at the beginning following the procedure 2,3 and is stored for the subsequent invoking. By inheriting the symbol used in Equation (A6), the  $j$ th equilibrium density on particle coordinate spatial mesh is denoted as  $n_{g0jh,p}$ . Then, the perturbative density on the particle coordinate spatial mesh contributed by the  $\mu_j$  component is

$$n_{g1jh,p} = n_{gjh,p} - n_{g0jh,p}. \quad (56)$$

5. The total perturbative density on the particle coordinate space computed by all  $j \in \{1, \dots, N_\mu\}$  is obtained by “MPI\_ALLREDUCE” the  $n_{g1j,hp}$  as  $\sum_j n_{g1j,hp} B_{hp} \delta\mu_j$ , where  $B_{hp}$  is the magnetic field magnitude at the node denoted by  $(h, p)$ .

For an initial equilibrium density profile on the gyrocenter coordinate frame shown by the purple line of Figure 1, assuming the equilibrium distribution of  $\mu$  obeying Equation (58), the density on the particle coordinate space after the gyroaverage operation is obtained and shown in Figure 1.

### 7.2.2. The Parallelization

MPI is used to the parallelization of the computing of  $\mu$ . The processors are divided into  $N_\mu$  sub-communicators.  $F_j(\mathbf{x}, \mu_j, U)$  with  $j \in \{1, \dots, N_\mu\}$  is exclusively computed by the  $j$ th sub-communicator. In addition, the respective precomputing matrixes of  $\Phi_j$  are stored in the  $j$ th sub-communicator.  $\tilde{\Phi}$  and  $n_{i1}(\mathbf{x})$  in Equations (23) and (17b) are computed by “MPI\_ALLREDUCE” the respective quantity stored in the processors of the same “color” with respect to the respective sub-communicator.

To calculate the advection of distribution function in the 4D domain  $(r, \Theta, x_\parallel, U)$ , two parallelization schemes are involved: the one for parallelizing  $x_\parallel$  with  $r, \Theta, U$  sequential is used to calculate the advection due to  $\dot{r}, \dot{\Theta}, \dot{U}$ ; the other one for parallelizing  $r, \Theta, U$  with  $x_\parallel$  sequential is used to compute the advection due to  $U$ . The 3D domain  $(r, \Theta, x_\parallel)$  with respect to the potential function and the density implements the scheme: the parallelization in  $x_\parallel$  with  $r, \Theta$  sequential is used to compute the characteristics  $\dot{r}, \dot{\Theta}, \dot{U}$  which are needed in computing the advection of the distribution in  $r, \Theta, U$  dimensions, and to solve QNE in the poloidal cross-section.

### 7.3. The Simulation Results

In the cylindrical coordinates system, the initial distribution is of the structure in Equation (14). The distribution function is specifically given as

$$F(0, r, \Theta, x_\parallel) = F_{eq}(r, \mu, U) \times \left( 1 + \eta \exp \left( -\frac{(r - r_p)^2}{\delta r} \right) \sum_{n_p, m_p} \cos \left( \frac{2\pi n_p}{L_\parallel} x_\parallel + m_p \Theta + \frac{2\pi p_p}{L_r} r \right) \right), \quad (57)$$

where the equilibrium function  $F_{eq}$  is

$$F_{eq}(r, \mu, U) = \frac{n_0(r) \exp \left( -\frac{U^2}{2T_i(r)} - \frac{\mu B}{T_i(r)} \right)}{(2\pi T_i(r))^{3/2}}, \quad (58)$$

where  $\eta$  is the parameter for the amplitude of the perturbation, and  $n_p, m_p, p_p$  are the respective mode numbers. The profile  $T_i(r)$ ,  $T_e(r)$  and  $n_0(r)$  are given by:

$$\mathcal{P}(r) = C_{\mathcal{P}} \exp \left( -\kappa_{\mathcal{P}} \delta r_{\mathcal{P}} \tanh \left( \frac{r - r_{\mathcal{P}}}{\delta r_{\mathcal{P}}} \right) \right) \quad (59)$$

where  $\mathcal{P} \in \{T_i, T_e, n_0\}$ ,  $C_{T_i} = C_{T_e} = 1$ , and

$$C_{n_0} = \frac{r_{\max} - r_{\min}}{\int_{r_{\min}}^{r_{\max}} \exp \left( -\kappa_{n_0} \delta r_{n_0} \tanh \left( \frac{r - r_{n_0}}{\delta r_{n_0}} \right) \right) dr}. \quad (60)$$

In Equation (59),  $C_{\mathcal{P}}$  is the normalized factor; the magnitude of  $\kappa_{\mathcal{P}}$  mainly determines the wide between  $\mathcal{P}_{\max}$  and  $\mathcal{P}_{\min}$ ; the magnitude of  $\delta r_{\mathcal{P}}$  determines mainly the steep of the gradient of the profile of  $\mathcal{P}$ .

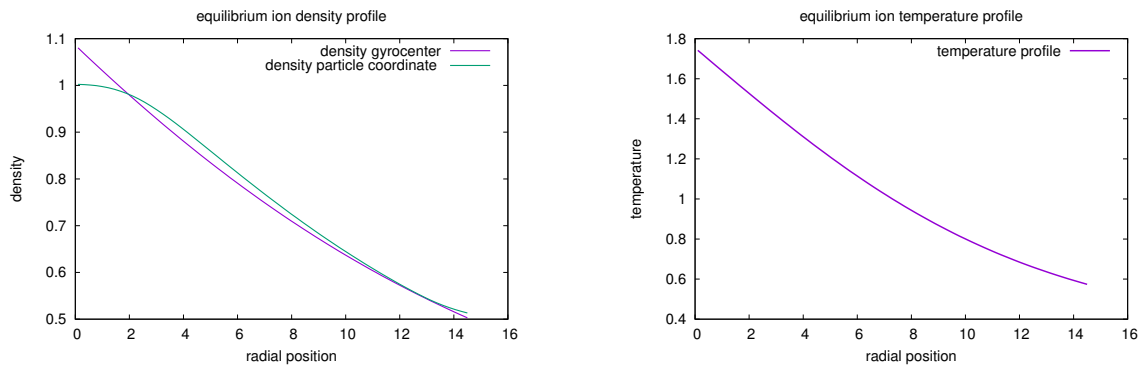
For all the following simulation cases, the time step  $\Delta t = 8$  is chosen and the data are stored every two time steps.

### 7.3.1. 1st Case: $n_p = 1, m_p = 0, p_p = 0$

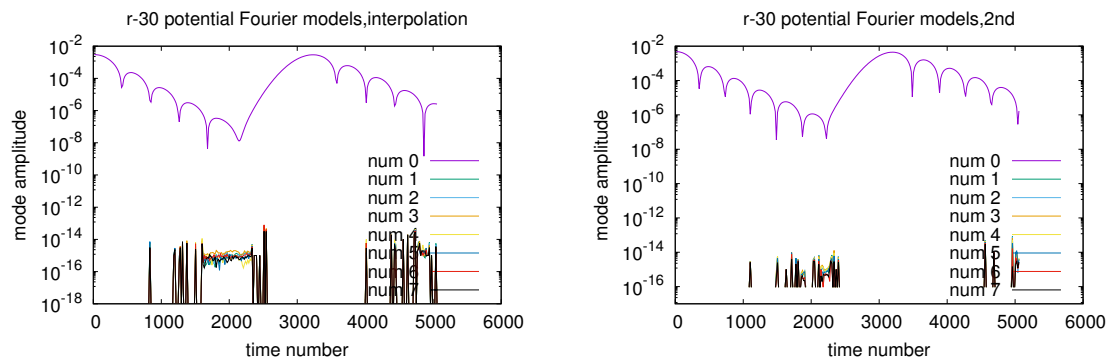
This case implements the perturbation not including the polar and radial modes. The used equilibrium profiles of ions' density and temperature are not steep in the radial dimension and are given in Figure 10. We consider the parameters:  $\eta = 10^{-4}$ ,  $\kappa_{n_0} = 13.0$ ,  $\kappa_{T_i} = \kappa_{T_e} = 20.0$ ,  $\delta r_{n_0} = 2.0$ ,  $\delta r_{T_i} = \delta r_{T_e} = 1.0$ ,  $L_{\parallel} = 1506.759067$ ,  $r_p = 0.45(r_{min} + r_{max})$ ,  $\delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}$ .

The simulated domain of  $r \times \Theta \times x_{\parallel} \times U \times \mu$  is chosen as  $(0.5\Delta r, 14.5) \times [0, 2\pi) \times (0, 1506.759067) \times (-7.32, 7.32) \times (0, 20)$ , where  $\Delta r$  is the length step in radial direction. This domain is divided into a mesh of  $64 \times 64 \times 32 \times 32 \times 63$  cells. The simulation is carried out by implementing 128 processes, which are executed by 4 nodes on ATLAS HPC of IRMA using hyperthreading. Each node has 24 CPUs and carries on 32 processes.

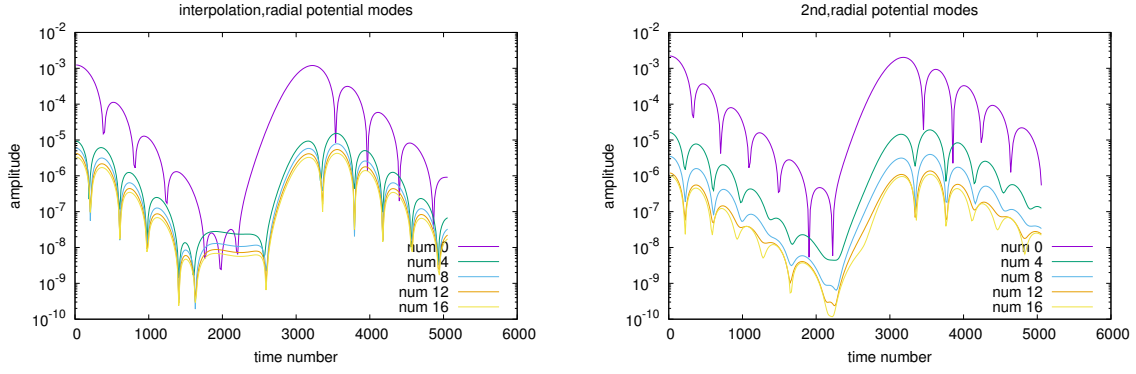
The evolution curves of the polar modes of the potential at the radial node 30 are shown in Figure 11. The radial spectrum evolution is shown in Figure 12. A good match of the results computed by the two solvers is revealed by these figures.



**Figure 10.** These figures are for the 1st case. The purple curve of the left figure is ion's equilibrium density profile on the gyrocenter coordinate frame, while the green curve is the ion's equilibrium density profile on the particle coordinate frame transformed from the purple curve by the gyroaverage operation. Ion's equilibrium temperature profile is given by the right figure.



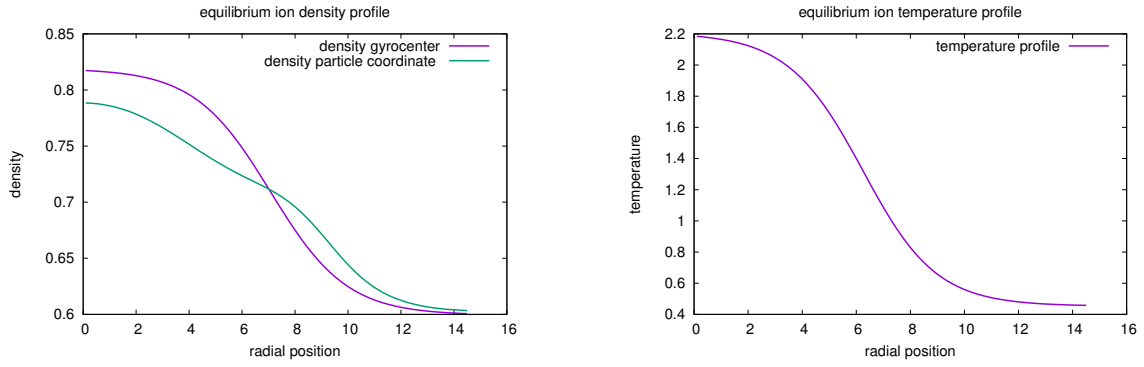
**Figure 11.** These figures are for the 1st case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.



**Figure 12.** These figures are for the 1st case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

### 7.3.2. 2nd Case: $n_p = 1, m_p = 15, p_p = 0$

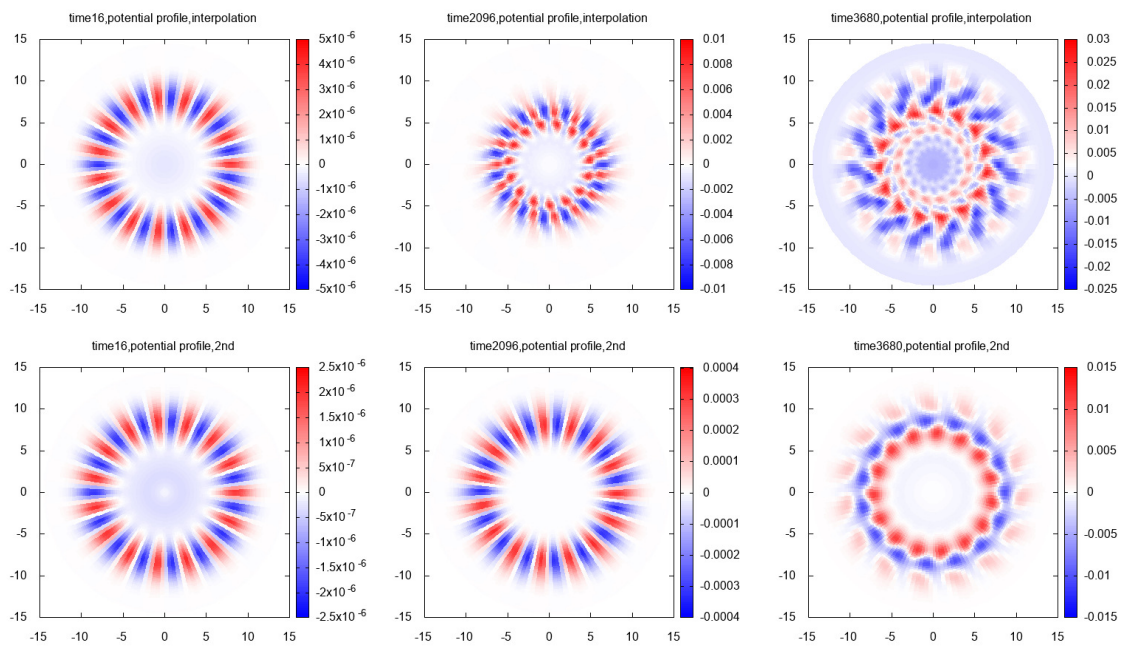
For this case, we could obtain  $k_{\Theta}\rho \approx 2$  at  $r = 7$  with  $\rho = 1$ . We consider the parameters:  $\eta = 10^{-4}$ ,  $\kappa_{n_0} = 13.0$ ,  $\kappa_{T_i} = \kappa_{T_e} = 66.0$ ,  $\delta r_{T_e} = \delta r_{T_i} = \delta r_{n_0} = 0.2$ ,  $L_{\parallel} = 1506.759067$ ,  $r_p = 0.45(r_{min} + r_{max})$ ,  $\delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}$ . The equilibrium profile of ion's density and temperature are given by Figure 13.



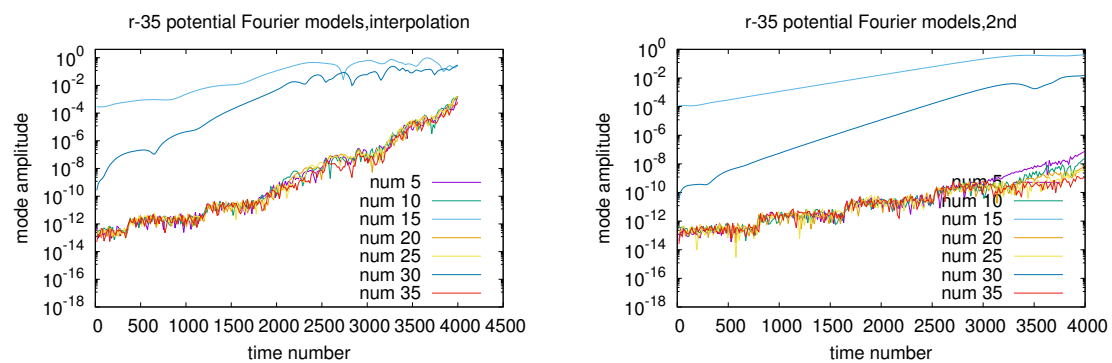
**Figure 13.** These figures are for the 2nd case. The purple curve of the left figure is ion's equilibrium density profile on the gyrocenter coordinate frame, while the green curve is on the particle coordinate frame transformed from the purple curve by the gyroaverage operation. Ion's equilibrium temperature profile is given by the right figure.

The simulated domain is the same with that of the 1st case and is divided into a mesh of  $64 \times 128 \times 32 \times 32 \times 63$  cells. For this case, the implementation of the processes and the partition of these processes on the nodes of ATLAS are the same with that of the 1st case.

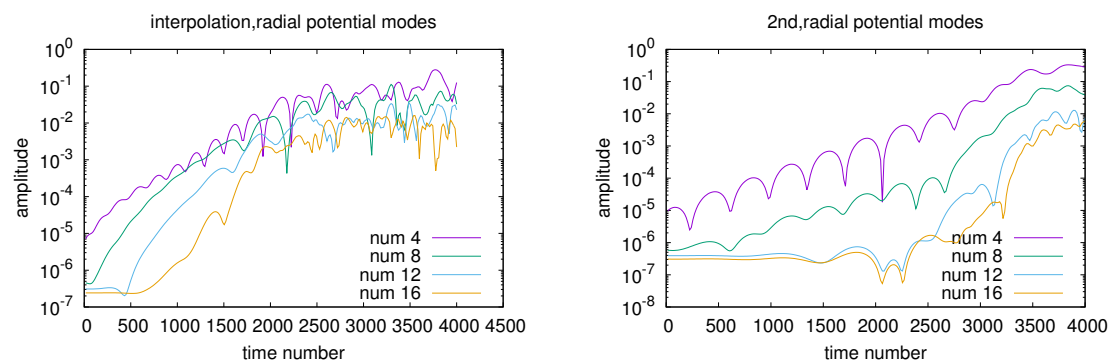
The potential profile on the polar cross-section at time moment 16, 2096, 3680 computed by the two solvers are shown in Figure 14. The ion temperature modes are driven by the equilibrium gradient. The evolution curves of the polar modes of the potential at the radial node 35 are presented in Figure 15, which shows that the growth rate computed by the interpolation algorithm is larger and the saturation time is earlier. The radial spectrum evolution is shown in Figure 16.



**Figure 14.** These figures are for the 2nd case. The upper row is the evolution of the potential profile over the polar cross-section computed by the interpolation algorithm, while the lower row is computed by the 2nd-order solver.



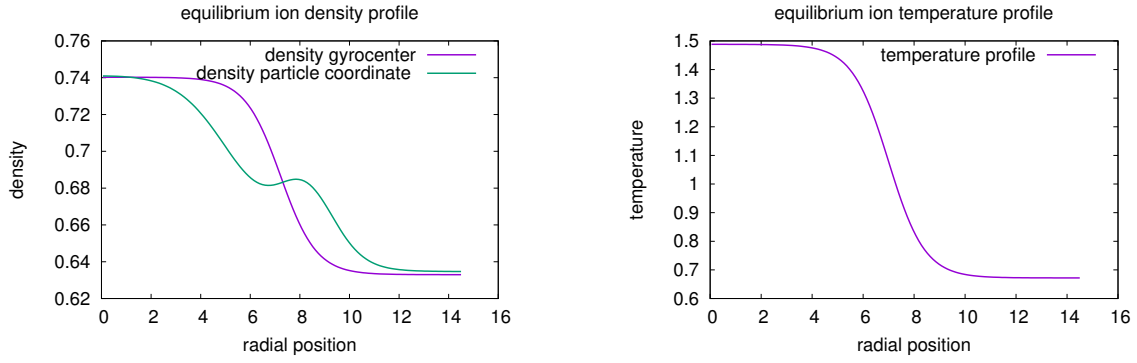
**Figure 15.** These figures are for the 2nd case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.



**Figure 16.** These figures are for the 2nd case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

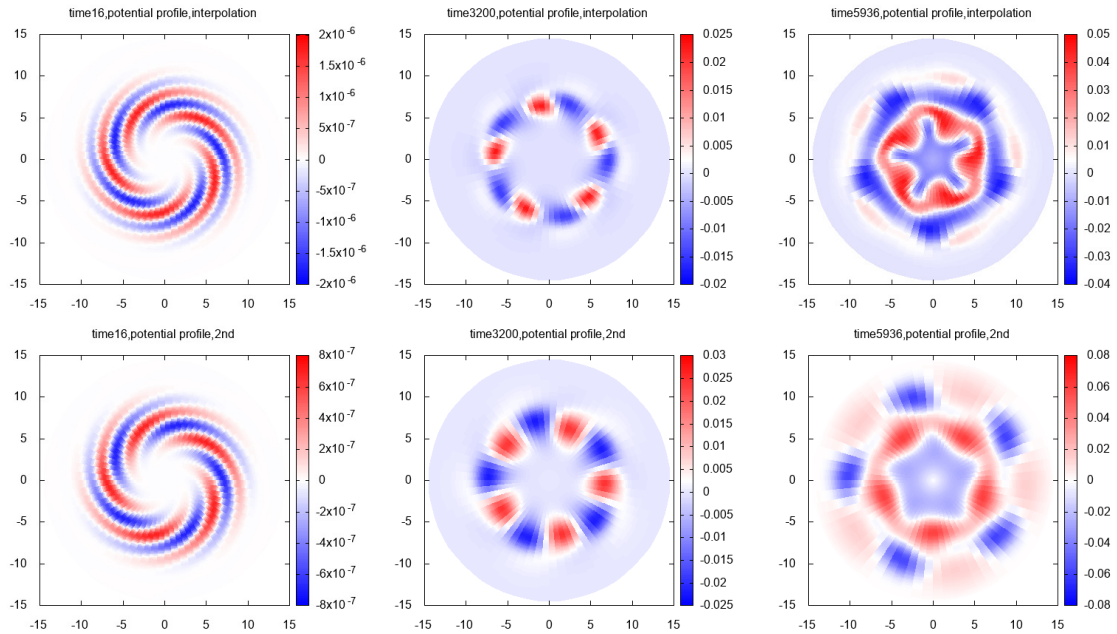
### 7.3.3. 3rd Case: $n_p = 1, m_p = 5, p_p = 7$

For this case, we could obtain  $k_r \rho \approx \pi$  with  $\rho = 1$ . We consider the parameters:  $\eta = 10^{-4}$ ,  $\kappa_{n_0} = 13.0$ ,  $\kappa_{T_i} = \kappa_{T_e} = 66.0$ ,  $\delta r_{T_e} = \delta r_{T_i} = \delta r_{n_0} = 0.1$ ,  $L_{||} = 1506.759067$ ,  $r_p = 0.45(r_{min} + r_{max})$ ,  $\delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}$ . The equilibrium profile of ion's density and temperature are given by Figure 17.

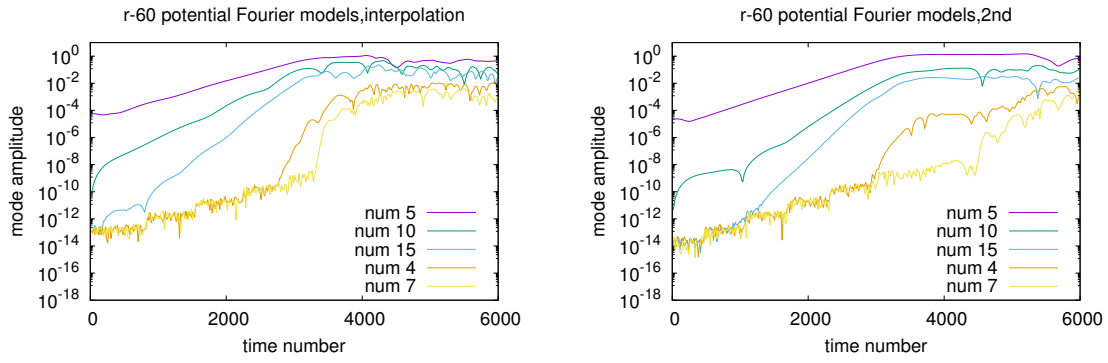


**Figure 17.** These figures are for the 3rd case. The purple curve of the left figure is ion's equilibrium density profile on the gyrocenter coordinate frame, while the green curve is on the particle coordinate frame transformed from the purple curve by the gyroaverage operation. Ion's equilibrium temperature profile is given by the right figure.

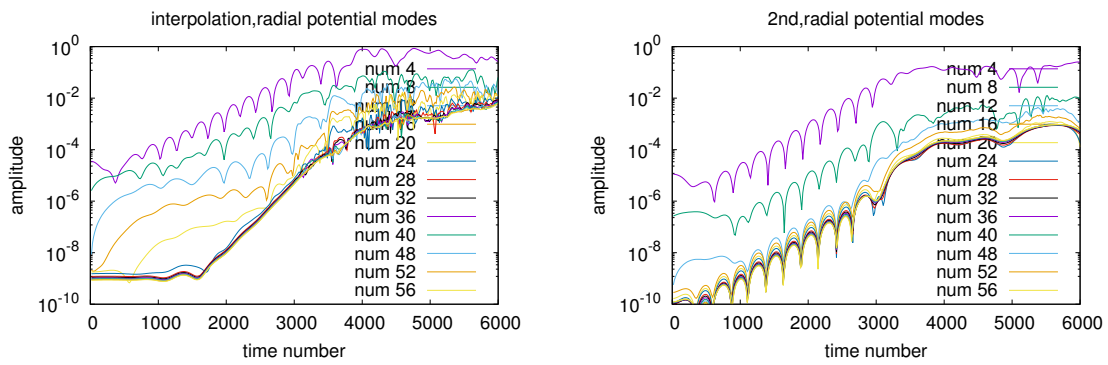
The simulated domain is the same and is divided into a mesh of  $128 \times 64 \times 32 \times 32 \times 63$  cells. The implementation of the processes on the nodes of ATLAS does not change. The potential profile on the polar cross-section at time moment 16, 3200, 5236 computed by the two solvers are shown in Figure 18. The evolution curves of the polar modes of the potential at the radial node 35 are presented in Figure 19. The radial spectrum evolution is given in Figure 20.



**Figure 18.** These figures are for the 3rd case. The cross-section profile of the potential computed by the interpolation solver and the 2nd-order solver.



**Figure 19.** These figures are for the 3rd case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

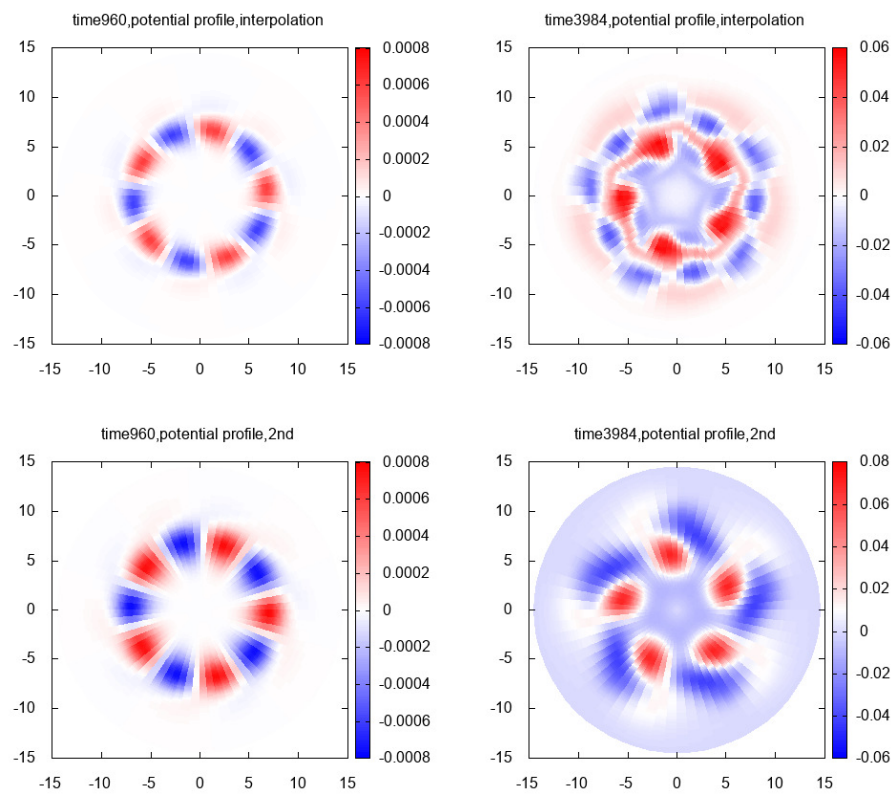


**Figure 20.** These figures are for the 3rd case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.

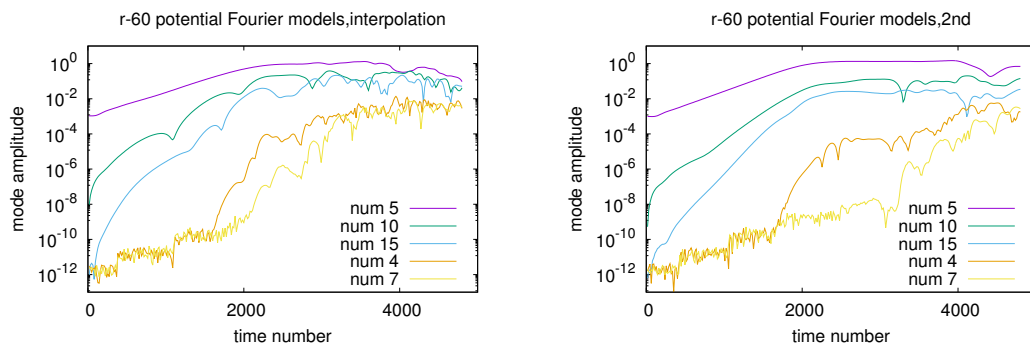
#### 7.3.4. 4th Case: $n_p = 1, m_p = 5, p_p = 0$

The parameters of this case are the same with those used in the 3rd case except that  $p_p = 0$  is chosen. The results are plotted in Figures 21, 22 and 23. Compared with the simulation results of the 3rd case, it can be observed from both figures that the growth rate of the polar Fourier modes without the initial radial wavelength are much larger than that with the presence of the initial radial wavelength. Such a fact may contribute a new element to the turbulence induced transport theory, which states that the shear flow could reduce the radial coherent length of the turbulent vortex, mitigating the transport rate as a consequence. Our simulations in 3rd and 4th cases suggest that with the presence of the low toroidal mode number, the short radial wavelength causes much smaller growth rate of the poloidal modes.

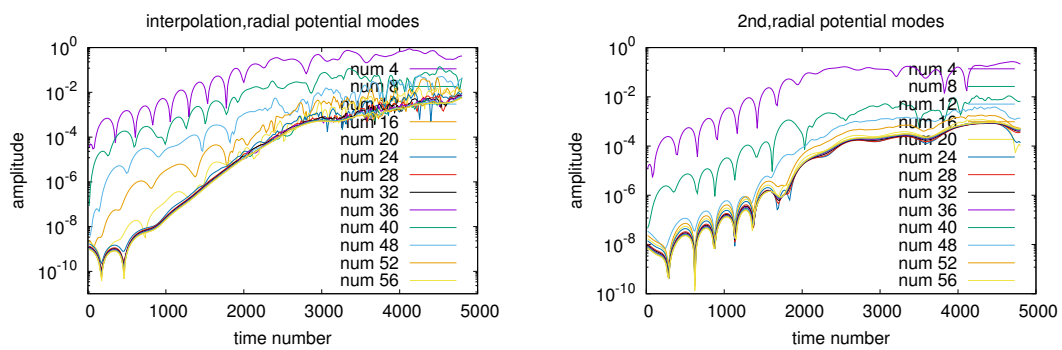




**Figure 21.** These figures are for the 4th case. The cross-section profile of the potential computed by the interpolation solver and the 2nd-order solver.



**Figure 22.** These figures are for the 4th case. The polar Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.



**Figure 23.** These figures are for the 4th case. The radial Fourier modes of the perturbative potential computed by the interpolation algorithm and the 2nd-order solver.



## 8. Conclusions and Discussion

This paper points out that the traditional low-order approximation of DGT is not suitable concerning the short-scale perturbation and the steep equilibrium profile. To obtain the more precise mesh-grids value of the function which is under the first gyroaverage or double gyroaverage in the presence of these extreme cases, the interpolation algorithm is developed by uniformly laying out many points on the Larmor circle surrounding the grid point, the function value at which is obtained by using the cubic spline piecewise basis function for the interpolation. Due to the periodic boundary condition of the cubic spline interpolation on the polar dimension, the interpolation coefficient matrix has the periodic block structure. Then, FFT can be implemented to speed up the process of obtaining the inverse of the discretized coefficient matrix of QNE. Since the density of each  $\mu_j$  depends on a weight factor  $\exp(-\frac{\mu B(\mathbf{x})}{T_e(\mathbf{x})})$  and all quantities as the functions of  $\mathbf{x}$  experience a transform  $\mathbf{x} \rightarrow \mathbf{x} - \rho_0(\mathbf{x}, \mu_j, \theta)$ , the Gaussian-Laguerre quadrature integrating method cannot be used and we developed an Euler–Maclaurin-based quadrature integrating method to obtain the quadrature integration for the distribution of  $\mu$ .

The interpolation algorithm itself and the interpolation solver for the numerical computation of QNE are benchmarked under various boundary conditions based on various analytical solutions. It is found that numerical solutions computed by both cases match the analytical solutions very well. The interpolation solver is also applied to the gyrokinetic integrated simulation which comprises the other Vlasov solver and the characteristic solver. The perturbations possessing various model numbers and the equilibrium profiles possessing various radial gradient are implemented in the integrated simulations. To observe the advantage of the interpolation algorithm, its numerical results are compared with those computed by the 2nd-order solver. When the equilibrium profile is not steep and the perturbation only has the non-zero mode number along the parallel spatial dimension, the results computed by the interpolation solver and the 2nd-order solver match each other well. When the gradient of the equilibrium profile is steep, the interpolation solver provides a bigger driving effect for the ion-temperature-gradient modes which possesses large polar mode numbers.

**Author Contributions:** Conceptualization, S.Z. and M.M.; Formal analysis, S.Z.; Investigation, S.Z. and M.M.; Methodology, S.Z., M.M. and C.S.; Software: S.Z., M.M. and C.S.; Project administration, M.M.; Validation, S.Z.; Writing & original draft, S.Z.; Writing & Review & Editing, S.Z. and M.M.

**Funding:** The first author was supported by ANR EXAMAG. This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work was granted access to the HPC resources of Aix-Marseille Université financed by the project Equip@Meso (ANR-10-EQPX-29-01) of the program “Investissements d’Avenir” supervised by the Agence Nationale de la Recherche

**Acknowledgments:** The first author appreciates the help of Sever Hirstoaga and Matthieu Boileau for the implementation of ATLAS HPC of IRMA.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. 2nd-Order Approximation of QNE

We first need to take Taylor expansion of DGT. According to Appendix B, the expansion rule of a general function  $\mathcal{A}(\mathbf{x} + \rho_0)$  is

$$\mathcal{A}(\mathbf{x} + \rho_0) = \sum_{n \geq 0} \frac{(\rho_0 \cdot \nabla)^n}{n!} \mathcal{A}(\mathbf{x}).$$

The gyroaverage of the following terms are needed

$$\begin{aligned}\langle (\boldsymbol{\rho}_0 \cdot \nabla) \mathcal{A}(\mathbf{x}) \rangle &= 0, \\ \langle (\boldsymbol{\rho}_0 \cdot \nabla)^2 \mathcal{A}(\mathbf{x}) \rangle &= \frac{1}{2} \rho_0^2 \nabla_{\perp}^2 \mathcal{A}(\mathbf{x}),\end{aligned}$$

where

$$\langle \mathcal{A}(\mathbf{x} + \boldsymbol{\rho}_0) \rangle \equiv \frac{1}{2\pi} \int \mathcal{A}(\mathbf{x} + \boldsymbol{\rho}_0) d\theta.$$

The following identity to reduce the perpendicular distribution to the density is also necessary

$$\int \rho_0^2 \partial_{\mu} F_{i0\perp} B d\mu = \frac{-1}{B\pi},$$

The first gyroaverage expanded up to the second order is

$$\Phi(\mathbf{x}, \mu) \equiv \langle \phi(\mathbf{x} + \boldsymbol{\rho}_0) \rangle \approx \left(1 + \frac{\rho_0^2 \nabla_{\perp}^2}{4}\right) \phi(\mathbf{x}). \quad (\text{A3})$$

The second gyroaverage expanded up to the second order is

$$\begin{aligned}\left(\frac{\Phi F_{i0}}{T_i}\right)(\mathbf{x} - \boldsymbol{\rho}_0, \mu_1, u_1) &\approx \left(1 + \frac{\rho_0^2 \nabla_{\perp}^2}{4}\right)^2 \left(\frac{F_{i0}}{T_i} \phi\right) \\ &\approx \frac{F_{i0} \phi}{T_i} + \frac{\rho_0^2 \nabla_{\perp}^2}{2} \left(\frac{F_{i0} \phi}{T_i}\right) \\ &\approx \frac{F_{i0} \phi}{T_i} + \frac{\rho_0^2 \nabla_{\perp} \phi}{2} \cdot \nabla_{\perp} \left(\frac{F_{i0}}{T_i}\right) + \frac{F_{i0}}{T_i} \frac{\rho_0^2 \nabla_{\perp}^2}{2} \phi\end{aligned} \quad (\text{A4})$$

The density derived from Equation (A4) is

$$2\pi \int \left\langle \left(\frac{\Phi F_{i0}}{T_i}\right)(\mathbf{x} - \boldsymbol{\rho}_0, \mu, U) \right\rangle B d\mu dU \approx \frac{n_0 \phi}{T_i} + \frac{\nabla_{\perp} \phi \cdot \nabla_{\perp} n_0}{B^2} + \frac{n_0}{B^2} \nabla_{\perp}^2 \phi. \quad (\text{A5})$$

Then, QNE derived by the 2nd-order approximation is

$$\frac{\phi}{T_i} - \frac{n_0 \phi}{n_{g0} T_i} - \frac{\nabla_{\perp} \phi \cdot \nabla_{\perp} n_0}{n_{g0} B^2} - \frac{n_0}{n_{g0} B^2} \nabla_{\perp}^2 \phi + \frac{\phi}{T_e} = \frac{n_{i1}}{n_{g0}}. \quad (\text{A6})$$

With the assumption of  $B = 1$ , Equation (A6) becomes

$$\frac{\phi}{T_i} - \frac{n_0 \phi}{n_{g0} T_i} - \frac{\nabla_{\perp} \phi \cdot \nabla_{\perp} n_0}{n_{g0}} - \frac{n_0}{n_{g0}} \nabla_{\perp}^2 \phi + \frac{\phi}{T_e} = \frac{n_{i1}}{n_{g0}}. \quad (\text{A7})$$

## Appendix B. The Examples of the Exact Solution of the Gyroaverage and Double-Gyroaverage Operation

We first consider a function of the form  $f(x + \epsilon g(x))$  depending on one scalar argument and a small parameter. What we are interested in is its expansion over  $\epsilon$ . The derivative of  $f(x + \epsilon g(x))$  over  $\epsilon$  at  $\epsilon = 0$  is derived as follows

$$d_{\epsilon} f(z)|_{\epsilon=0} = d_{\epsilon} z \partial_z f(z)|_{\epsilon=0} = g(x) \partial_x f(x), z \equiv x + \epsilon g(x), \quad (\text{A8})$$

where  $d_{\epsilon} \equiv d/d\epsilon$ . The second order derivative of  $f(z)$  over  $\epsilon$  at  $\epsilon = 0$  is

$$\begin{aligned}d_{\epsilon}(d_{\epsilon} f(z))|_{\epsilon=0} &= d_{\epsilon}(d_{\epsilon} z \partial_z f(z))|_{\epsilon=0} = (d_{\epsilon}(d_{\epsilon} z)) \partial_z f(z)|_{\epsilon=0} + d_{\epsilon} z (d_{\epsilon} \partial_z f(z))|_{\epsilon=0} \\ &= d_{\epsilon} g(x)|_{\epsilon=0} + (d_{\epsilon} z)(d_{\epsilon} z) \partial_z \partial_z f(z)|_{\epsilon=0} = g^2(x) \partial_z^2 f(z).\end{aligned}$$

The second order derivative of  $f(z)$  over  $\epsilon$  at  $\epsilon = 0$  is

$$d_{\epsilon}^2 f(x + \epsilon g(x)) \Big|_{\epsilon=0} = g^2(x) \partial_x^2 f(x).$$

It is easy to derive that the  $n$ -th derivative of  $f(x + \epsilon g(x))$  over  $\epsilon$  at  $\epsilon = 0$  is

$$d_{\epsilon}^n f(x + \epsilon g(x)) \Big|_{\epsilon=0} = g^n(x) \partial_x^n f(x). \quad (\text{A9})$$

Then, the Taylor expansion of  $f(x + \epsilon g(x))$  over  $\epsilon$  is

$$f(x + \epsilon g(x)) = \sum_{n \geq 0} \frac{\epsilon^n}{n!} g^n(x) \partial_x^n f(x). \quad (\text{A10})$$

Now we change  $g(x)$  to be the vector  $\rho_0$ . In Cartesian coordinate frame,  $\rho_0(\mathbf{x}) \cdot \nabla$  can be written as

$$\rho_0(\mathbf{x}) \cdot \nabla = \sum_i \rho_i(\mathbf{x}) \partial_{x_i},$$

Then, Equation (A10) is changed to be

$$f(\mathbf{x} + \rho_0(\mathbf{x})) = \sum_{n \geq 0} \frac{1}{n!} \left( \sum_i \rho_i(\mathbf{x}) \partial'_{x_i} \right)^n f(\mathbf{x}) = \exp \left( \sum_i \rho_i(\mathbf{x}) \partial'_{x_i} \right) f(\mathbf{x}), \quad (\text{A11})$$

where  $'$  means  $\partial'_{x_i}$  does not operate on any  $\rho_i$ s.

In Cartesian coordinate frame, we have

$$\rho_0 \cdot \nabla = \rho \cos \alpha_1 \partial'_{x_1} + \rho \cos(\alpha_1 - \pi/2) \partial'_{x_2},$$

where  $\alpha_1$  is the angle between the Larmor radius vector and  $\mathbf{e}_1$  a unit vector. For

$$f(\mathbf{x}) = \exp(inx_1 + imx_2),$$

we have

$$\begin{aligned} \mathcal{J}_1(\mu) f(\mathbf{x}) &= \langle \exp(in\rho \cos \alpha_1 + im\rho \cos(\alpha_1 - \pi/2)) \rangle \exp(inx_1 + imx_2) \\ &= \left\langle \exp(i\rho \sqrt{n^2 + m^2} \cos(\alpha_1 + \beta + \pi/2)) \right\rangle \exp(inx_1 + imx_2) \\ &= \left\langle \sum_{q=0} i^q J_q(\rho \sqrt{n^2 + m^2}) \exp(iq(\alpha_1 + \beta + \pi/2)) \right\rangle \exp(inx_1 + imx_2) \\ &= J_0(\rho \sqrt{n^2 + m^2}) \exp(inx_1 + imx_2), \end{aligned} \quad (\text{A12})$$

where  $\cos \beta = n / \sqrt{n^2 + m^2}$ . With the same reason, we can derive

$$\mathcal{J}_2(\mu) \mathcal{J}_1(\mu) f(\mathbf{x}) = J_0^2(\rho \sqrt{n^2 + m^2}) \exp(inx_1 + imx_2) \quad (\text{A13})$$

For  $f_1(\mathbf{x}) = x_2$ , it is easy to obtain the following results

$$\mathcal{J}_1(\mu) x_2 = x_2, \mathcal{J}_2(\mu) \mathcal{J}_1(\mu) x_2 = x_2$$

## References

1. Diamond, P.H.; Itoh, S.I.; Itoh, K.; Hahm, T.S. Zonal flows in plasma—A review. *Plasma Phys. Control. Fusion* **2005**, *47*, R35. [\[CrossRef\]](#)
2. Lin, Z.; Hahm, T.S.; Lee, W.W.; Tang, W.M.; White, R.B. Turbulent Transport Reduction by Zonal Flows: Massively Parallel Simulations. *Science* **1998**, *281*, 1835–1837. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Biglari, H.; Diamond, P.H.; Terry, P.W. Influence of sheared poloidal rotation on edge turbulence. *Phys. Fluids B Plasma Phys.* **1990**, *2*, 1. [\[CrossRef\]](#)
4. Wesson, J. *Tokamaks*, 3rd ed.; CLARENDON Press: Oxford, UK, 2004.
5. Chen, F. *Introduction to Plasma Physics and Controlled Fusion*; Springer: New York, NY, USA, 2010.
6. Hazeltine, R.D.; Meiss, J.D. *Plasma Confinement*; Addison-Wesley, Advanced Book Program: Boston, MA, USA, 1992.
7. White, R.B. *Theory of Tokamak Plasmas*, 1st ed.; Elsevier: Amsterdam, The Netherlands, 1989.
8. Zohm, H. Edge localized modes (ELMs). *Plasma Phys. Control. Fusion* **1996**, *38*, 105. [\[CrossRef\]](#)
9. Hayashi, N.; Takizuka, T.; Aiba, N.; Oyama, N.; Ozeki, T.; Wiesen, S.; Parail, V. Integrated simulation of ELM energy loss and cycle in improved H-mode plasmas. *Nucl. Fusion* **2009**, *49*, 95015. [\[CrossRef\]](#)
10. Burrell, K.H.; West, W.P.; Doyle, E.J.; Austin, M.E.; Casper, T.A.; Gohil, P.; Greenfield, C.M.; Groebner, R.J.; Hyatt, A.W.; Jayakumar, R.J.; et al. Advances in understanding quiescent H-mode plasmas in DIII-D. *Phys. Plasmas* **2005**, *12*, 56121. [\[CrossRef\]](#)
11. Hubbard, A.E. Physics and scaling of the H-mode pedestal. *Plasma Phys. Control. Fusion* **2000**, *42*, A15. [\[CrossRef\]](#)
12. Lönnroth, J.S.; Parail, V.; Dnestrovskij, A.; Figarella, C.; Garbet, X.; Wilson, H.; Contributors, J.E. Predictive transport modelling of type I ELMy H-mode dynamics using a theory-motivated combined ballooning–peeling model. *Plasma Phys. Control. Fusion* **2004**, *46*, 1197. [\[CrossRef\]](#)
13. Ryter, F.; Angioni, C.; Beurskens, M.; Cirant, S.; Hoang, G.T.; Hogeweij, G.M.D.; Imbeaux, F.; Jacchia, A.; Mantica, P.; et al. Experimental studies of electron transport. *Plasma Phys. Control. Fusion* **2001**, *43*, A323. [\[CrossRef\]](#)
14. Suttrop, W. The physics of large and small edge localized modes. *Plasma Phys. Control. Fusion* **2000**, *42*, A1. [\[CrossRef\]](#)
15. Dickinson, D.; Saarelma, S.; Scannell, R.; Kirk, A.; Roach, C.M.; Wilson, H.R. Towards the construction of a model to describe the inter-ELM evolution of the pedestal on MAST. *Plasma Phys. Control. Fusion* **2011**, *53*, 115010. [\[CrossRef\]](#)
16. Garbet, X.; Idomura, Y.; Villard, L.; Watanabe, T.H. Gyrokinetic simulations of turbulent transport. *Nucl. Fusion* **2010**, *50*, 43002. [\[CrossRef\]](#)
17. Lee, W.W. Gyrokinetic particle simulation model. *J. Comput. Phys.* **1987**, *72*, 243. [\[CrossRef\]](#)
18. Lee, W.W. Gyrokinetic approach in particle simulation. *Phys. Fluids* **1983**, *26*, 556–562. [\[CrossRef\]](#)
19. Dubin, D.H.E.; Krommes, J.A.; Oberman, C.; Lee, W.W. Nonlinear gyrokinetic equations. *Phys. Fluids* **1983**, *26*, 3524–3535. [\[CrossRef\]](#)
20. Lin, Z.; Tang, W.M.; Lee, W.W. Gyrokinetic particle simulation of neoclassical transport. *Phys. Plasmas* **1995**, *2*, 2975–2988. [\[CrossRef\]](#)
21. Wang, W.X.; Lin, Z.; Tang, W.M.; Lee, W.W.; Ethier, S.; Lewandowski, J.L.V.; Rewoldt, G.; Hahm, T.S.; Manickam, J. Gyro-kinetic simulation of global turbulent transport properties in tokamak experiments. *Phys. Plasmas* **2006**, *13*, 92505. [\[CrossRef\]](#)
22. Jenko, F.; Dorland, W.; Kotschenreuther, M.; Rogers, B.N. Electron temperature gradient driven turbulence. *Phys. Plasmas* **2000**, *7*, 1904–1910. [\[CrossRef\]](#)
23. Parker, S.E.; Kim, C.; Chen, Y. Large-scale gyrokinetic turbulence simulations: Effects of profile variation. *Phys. Plasmas* **1999**, *6*, 1709–1716. [\[CrossRef\]](#)
24. Candy, J.; Waltz, R.E. An Eulerian gyrokinetic-Maxwell solver. *J. Comput. Phys.* **2003**, *186*, 545–581. [\[CrossRef\]](#)
25. Chang, C.S.; Ku, S.; Diamond, P.; Adams, M.; Barreto, R.; Chen, Y.; Cummings, J.; D’Azevedo, E.; Dif-Pradalier, G.; Ethier, S.; et al. Whole-volume integrated gyrokinetic simulation of plasma turbulence in realistic diverted-tokamak geometry. *J. Phys. Conf. Ser.* **2009**, *180*, 12057. [\[CrossRef\]](#)

26. Idomura, Y.; Ida, M.; Kano, T.; Aiba, N.; Tokuda, S. Conservative global gyrokinetic toroidal full-f five-dimensional Vlasov simulation. *Comput. Phys. Commun.* **2008**, *179*, 391–403. [\[CrossRef\]](#)
27. Chen, Y.; Parker, S.E. A  $\delta f$  particle method for gyrokinetic simulations with kinetic electrons and electromagnetic perturbations. *J. Comput. Phys.* **2003**, *189*, 463–475. [\[CrossRef\]](#)
28. Gorler, T.; Lapillonne, X.; Brunner, S.; Dannert, T.; Jenko, F.; Merz, F.; Told, D. The global version of the gyrokinetic turbulence code GENE. *J. Comput. Phys.* **2011**, *230*, 7053–7071. [\[CrossRef\]](#)
29. Brizard, A.J. Nonlinear Gyrokinetic Tokamak Physics. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1990.
30. Brizard, A.J.; Hahm, T.S. Foundations of nonlinear gyrokinetic theory. *Rev. Mod. Phys.* **2007**, *79*, 421–468. [\[CrossRef\]](#)
31. Robert, G.M. Toeplitz and Circulant Matrices: A Review. *Found. Trends® Commun. Inf. Theory* **2006**, *2*, 155–239
32. SELALIB. Available online: <http://selalib.gforge.inria.fr/> (accessed on 4 December 2017).
33. Crouseilles, N.; Glanc, P.; Hirstoaga, S.A.; Madaule, E.; Mehrenberger, M.; Pétri, J. A new fully two-dimensional conservative semi-Lagrangian method: Applications on polar grids, from diocotron instability to ITG turbulence. *Eur. Phys. J. D* **2014**, *68*, 252. [\[CrossRef\]](#)
34. Sonnendrücker, E.; Roche, J.; Bertrand, P.; Ghizzo, A. The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation. *J. Comput. Phys.* **1999**, *149*, 201–220. [\[CrossRef\]](#)
35. Grandgirard, V.; Brunetti, M.; Bertrand, P.; Besse, N.; Garbet, X.; Ghendrih, P.; Manfredi, G.; Sarazin, Y.; Sauter, O.; Sonnendrücker, E.; et al. A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation. *J. Comput. Phys.* **2006**, *217*, 395–423. [\[CrossRef\]](#)
36. Grandgirard, V.; Sarazin, Y.; Garbet, X.; Dif-Pradalier, G.; Ghendrih, P.; Crouseilles, N.; Latu, G.; Sonnendrücker, E.; Besse, N.; Bertrand, P. Computing ITG turbulence with a full-f semi-Lagrangian code. *Commun. Nonlinear Sci. Numer. Simul.* **2008**, *13*, 81–87. [\[CrossRef\]](#)
37. Latu, G.; Mehrenberger, M.; Güçlü, Y.; Ottaviani, M.; Sonnendrücker, E. Field-Aligned Interpolation for Semi-Lagrangian Gyrokinetic Simulations. *J. Sci. Comput.* **2017**. [\[CrossRef\]](#)
38. Steiner, C.; Mehrenberger, M.; Crouseilles, N.; Grandgirard, V.; Latu, G.; Rozar, F. Gyroaverage operator for a polar mesh. *Eur. Phys. J. D* **2015**, *69*, 18. [\[CrossRef\]](#)
39. Steiner, C.; Mehrenberger, M.; Crouseilles, N.; Philippe, H. Quasi-neutrality equation in a polar mesh. Available online: <https://hal.archives-ouvertes.fr/hal-01248179/document> (accessed on 4 November 2015).
40. Coulette, D.; Besse, N. Numerical comparisons of gyrokinetic multi-water-bag models. *J. Comput. Phys.* **2013**, *248*, 1–32. [\[CrossRef\]](#)



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).